UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Testing dynamic routing protocols with VNX virtual networks

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Yeray Sainz Lorenzo

In partial fulfilment

of the requirements for the degree in

Telematics Engineering

Advisor: Jose Luis Muñoz Tapia

Barcelona, January 2017

Abstract

The developed project consists in implementing and testing routing protocols in virtual networks using an open-source virtualization tool (VNX). With this tool, several virtual machines running Linux operating system can be started and virtually connected to each other. Within each virtual machine, Quagga network routing software suite is used to implement the routing protocols used.

The routing protocol that has been widely tested in this project is the Border Gateway Protocol (BGP). BGP is used by most Internet service providers to exchange a great amount of routes. The implementation of BGP in Quagga has been successfully tested in the virtual scenarios, and the results show that the protocol behaviour is the expected. Additionally the Intermediate System to Intermediate System (IS-IS) protocol has been tested, but not exhaustively because of the existence of bugs in the Quagga implementation.

Resum

El projecte desenvolupat ha consistit en implementar i provar protocols d'enrutament en xarxes virtuals, fent servir l'eina de software lliure VNX. Amb aquesta eina, una sèrie de màquines virtuals amb sistema operatiu Linux es poden posar en marxa i connectar-se virtualment entre elles. A cada màquina virtual, s'utilitza el software d' enrutament Quagga per a implementar els protocols d' enrutament utilitzats.

El protocol d' enrutament que ha sigut àmpliament provat en aquest projecte és el Border Gateway Protocol (BGP). El BGP és fet servir per la majoría de proveïdors d' Internet, per a intercanviar grans volums de rutes. La implementació del protocol BGP ha estat provada satisfactòriament als escenaris virtuals, i els resultats mostren que el comportament del protocol és l'esperat. Addicionalment, s'ha provat el protocol Intermediate System to Intermediate System (IS-IS), però d'una forma limitada degut a l'existència de "bugs" a l' implementació en Quagga.

Resumen

El proyecto desarrollado ha consistido en implementar y probar protocolos de enrutamiento en redes virtuales, usando la herramienta de software libre VNX. Con esta herramienta, una serie de máquinas virtuales con sistema operativo Linux se pueden poner en marcha y conectarse entre sí virtualmente. En cada máquina virtual se utiliza el software de enrutamiento Quagga, para implementar los protocolos de enrutamiento utilizados.

El protocolo de enrutamiento que ha sido ampliamente probado en este proyecto es el Border Gateway Protocol (BGP). El BGP es utilizado por la mayoría de proveedores de Internet, para intercambiar grandes volúmenes de rutas. La implementación de el protocolo BGP ha estado probada satisfactoriamente en los escenarios virtuales, y los resultados muestran que el comportamiento del protocolo es el esperado. Adicionalmente, se ha probado el protocolo Intermediate System to Intermediate System (IS-IS), pero de una forma limitada debido a la existencia de "bugs" en la implementación de Quagga.

Acknowledgements

I would like to express my gratitude to this project advisor, Jose Luis Muñoz Tapia, for having proposed this thesis' topic, as well as for keeping a periodical tracking of the thesis and suggesting ideas to improve it.

Revision history and approval record

Revision	Date	Purpose
0	06/01/2017	Document creation
1	11/01/2017	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Yeray Sainz Lorenzo	yeray.sainz.93@gmail.com
Jose Luis Muñoz Tapia	jose.munoz@entel.upc.edu

Written by: Yeray Sainz Lorenzo		Reviewed and approved by: Jose Luis Muñoz Tapia	
Date	06/01/2017	Date	15/01/2017
Name	Yeray Sainz Lorenzo	Name	Jose Luis Muñoz Tapia
Position	Project Author	Position	Project Supervisor

Table of contents

The table of contents must be detailed. Each chapter and main section in the thesis must be listed in the “Table of Contents” and each must be given a page number for the location of a particular text.

1. Introduction

In the introductory section an overview of the project is explained, including: Objectives, requirements and specifications, methods and procedures, work plan, deviations from the initial plan and incidences.

1.1. Objectives

The purpose of this project has been testing dynamic routing protocols using a virtualization tool, Virtual Networks over Linux, and a routing software suite for Linux systems, Quagga.

The scenarios developed, as well as being used to test the protocol in the virtual networks, has been used to develop teaching practices.

The objectives can be summarized into the following points:

- Routing protocols documentation development.
- Filesystem configuration to suit the scenarios requirements.
- Scenarios development / Network design.
- Routing protocols testing.
- Practices format to test the protocols.

1.2. Requirements and Specifications

The requirements of this project can be separated into two groups:

- Theoretical requirements:
  - Knowledge of the routing protocols used.
  - VNX language (XML based).
- Practical requirements:
  - Use of Linux.
  - LXC virtual containers (virtual machines in the scenarios).
  - Use of Quagga routing software.
  - Use of the VNX virtualization tool.

1.3. Methods and Procedures review

The following procedures have been done during the project:

- Studying routing protocols.
- Develop documentation about the routing protocols.
- Installing and testing the VNX virtualization tool.
- Tune and test virtual machines (LXC).
- Develop scenarios.
- Test protocols in the scenarios.
- Develop practices from the scenarios.

The first step has been studying the BGP routing protocol from different sources. Then a documentation that includes theoretical examples has been developed (Appendix A). The next step has been the preparation to simulate the virtual networks. This preparation includes the installation of the VNX virtualization, and learning how to create scenarios using its XML based language.

Once VNX has been tested, LXC software is installed to manage and configure the virtual container file system that will be used in the scenarios' virtual machines.

Once the file system is tuned so that it has Quagga installed, and the BGP daemon running, the next step is developing the VNX scenarios. Once the scenarios and its figures are documented, scenarios are started using VNX and configured in different ways to test different aspects if the protocol.

To conclude, practices documentation is developed from the different behaviours observed.

1.4. Third-party resources

Even though this thesis is not a continuation of another thesis, the following parts have developed the software used to create the virtual networks scenarios and test the routing protocols:

- VNX: Telematics Engineering Department from the Technical University of Madrid
- LXC containers: Virtuozzo, IBM, Google, and other personal developers.

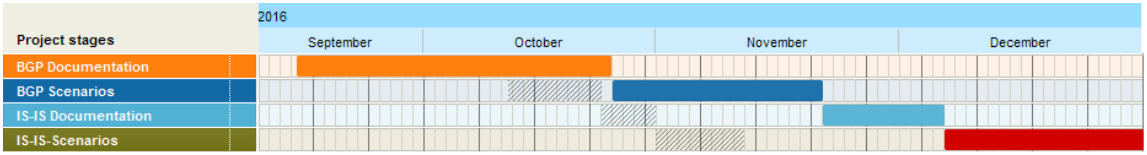
1.5. Work Plan

Project: Testing dynamic routing protocols with VNX virtual networks	WP ref: 1	
Major constituent: BGP documentation development		
Short description: Making documentation about the BGP protocol	Planned start date: 15/09/2016 Planned end date: 25/10/2016	
	Start event: 15/09/2016 End event: 19/10/2016	
	Deliverables:	Dates:

Project: Testing dynamic routing protocols with VNX virtual networks	WP ref: 2	
Major constituent: Developing BGP scenarios + practices		
Short description: Developing scenarios using LXC virtual machines, ubuntu 16.04 OS, and VNX software.	Planned start date: 26/10/2016 Planned end date: 21/11/2016	
	Start event: 20/10/2016 End event: 17/11/2016	
	Deliverables:	Dates:

Project: Testing dynamic routing protocols with VNX virtual networks	WP ref: 3	
Major constituent: IS-IS documentation development		
Short description: Making documentation about the IS-IS protocol.	Planned start date: 22/11/2016 Planned end date: 6/12/2016	
	Start event: 18/11/2016 End event:20/12/2016	
	Deliverables:	Dates:

Project: Testing dynamic routing protocols with VNX virtual networks	WP ref: 4	
Major constituent: Developing IS-IS scenarios + practices		
Short description: Developing scenarios using lxc virtual machines, ubuntu 16.04 OS, and VNX software.	Planned start date: 07/12/2016 Planned end date: 20/12/2016	
	Start event: 21/12/2016 End event: 4/1/2017	
	Deliverables:	Dates:



1.6. Deviations from the original plan

The original plan was developing MPLS scenarios as additional content. Since there was not a Quagga implementation for the MPLS protocol, IS-IS was chosen. Nevertheless, BGP testing has been the target of this project since the beginning, and it has been successfully tested.

2. State of the art of the technology used in this thesis:

2.1. Software used

The software used in this project consists in the following parts:

- Virtual Networks over Linux (VNX)
- Linux Containers (LXC)
- Quagga routing software suite

2.1.1. VNX

Virtual Networks over Linux (VNX) is an open-source virtualization tool, that allows creating virtual scenarios of virtual machines. The virtual nodes and networks are defined by the user, using an XML based language. By using VNX, it is possible to simulate networks and avoiding the investment and management costs that would suppose testing the networks in real equipment. Once the scenario is defined in an XML file, VNX parses the code and automatically builds the virtual scenario over a Linux machine.

Once of the improvements that VNX includes, and its predecessor (VNUML) did not, is the capability to use LXC containers as virtual machines.

2.1.2. LXC

Linux Containers (LXC) is an operating-system-level virtualization tool, to run Linux system over a single Linux kernel from a Linux host. LXC provide cgroups functionality, which isolate the resource usage of the virtual containers, and isolated namespaces, which provide a virtual isolated environment for processes.

In this thesis, the operating system running in the virtual containers is the same that runs in the physical host, to avoid possible incompatibilities. The operative system used is Ubuntu 16.04.

2.1.3. Quagga

The routing software suite used in the virtual containers is Quagga. This software for Unix platforms provides implementation of the most common routing protocols, such as RIP, OSPF, BGP and IS-IS. Each routing protocol has its own daemon, which can be all managed from a cohesive front-end command line interface called "vtysh". The core daemon, called "zebra", is an abstraction level of the Linux kernel.

2.2. Routing Protocols Implemented

The project has been focused in the BGP protocol and, to a lesser extent, the IS-IS protocol. In the following lines, the main features of these two protocols are about to be explained.

2.2.1. Border Gateway Protocol

The Border Gateway Protocol (BGP) is considered the standard Exterior Gateway Protocol (EGP). Being an EGP means that it is a routing protocol used between different administrative routing domains, for instance, between two different Internet service providers. The opposite of an EGP is an Interior Gateway Protocol (IGP), which is a type of protocol that usually runs within the same administrative domain.

BGP is transmits a route or set of routes (also called prefixes), in its route announcements. These announcements contain a path vector, that indicates the Autonomous Systems (AS) a certain route or group of routes has traversed. An AS is a group of routers, usually belonging to the same administrative domain. The path vector containing the AS list is used to avoid routing information loops within different AS.

BGP is focused on processing routing information properly, and features such as error control and flow control rely on TCP. That is, when two BGP routers exchange information between each other, the information in exchanged over a TCP connection. Another remarkable feature in BGP is that forwarding decisions are based in policies implemented in its routers. The policies can be taken from the values of the attributes that are carried in the route announcements between routers. For instance, an attribute could be modified when announcing a route to a neighbour AS that is linked to our AS through two different links, in order to influence the inbound traffic to our AS.

The main differences between BGP and an IGP, are the following ones:

- BGP is focused on processing routing information and announcing a great volume of network reachability information, while IGPs focus in convergence speed.

- On one hand, policies play an important role in BGP, since are used to make routing decisions. On the other hand, IGPs do not need policy information propagations, which make them suitable to run within an AS.

- Since IGPs carry topology information, this feature limits its scalability. BGP does not carry topology information, which makes it far more scalable.

A complete documentation about BGP can be found in Appendix A.

2.2.2. Intermediate System to Intermediate System

Intermediate System to Intermediate System (IS-IS) protocol is a link-state protocol that is used as an IGP, for instance, within a BGP AS. The main difference between IS-IS, and other IGPs, is that it runs in the layer 2 of the OSI protocol stack. This fact, makes IS-IS use its own jargon, whose details can be checked in the Appendix B, which contains detailed documentation about IS-IS. Another property of running on layer 2, is that it can carry other protocols over it, such as IP.

The IS-IS topology is separated into two topologies. The Level-1 topology is made from Level-1 IS-IS routers and links, which make an IS-IS area. The Level-2 topology is made from routers belonging to the Level-2 topology, or belonging to the Level-1 Level-2 topology. Within an area, a router that forwards traffic to other areas belongs to both levels. Topology information is leaked from Level-1 to Level-2, but not in the opposite way. Level-2 topology could be considered as the backbone of a network running IS-IS, since it takes care of the inter-area routing, as well as containing the topology information of all the Level-1 areas.

A remarkable feature, which makes IS-IS a good scalable IGP, is that it can include within a unique packet the link-state information of its links.

More precise information about the IS-IS protocol is located in Appendix B.

3. Project Development and Methodology:

The first step has been studying and developing documentation about BGP. The reference books to learn BGP are included in the bibliography section.

Once the documentation has been developed, the next step has been the installation and configuration of the virtualization tool VNX, the LXC's file system and Quagga routing software. Once these steps were finished, additional IS-IS documentation was done, and tested in VNX.

3.1. Theoretical Study

The knowledge about BGP has been acquired mainly from books [1] and [2], in the bibliography section, as well as other resources from the Internet.

Regarding IS-IS, book [3] in the bibliography section has been used

3.2. Software Installation and Configuration

3.2.1. VNX Installation

From the VNX website [3], following the installation guide, the installation has been successful and without errors through the command line interface in the physical host. The next step has been learning how to design the topology of the networks, by checking different sample scenarios' configuration files, as well as the tags' description contained in the website's documentation section.

3.2.2. LXC Installation

Firstly, the software used to run and manipulate LXC virtual machines is downloaded and installed from the LXC support website [4].

The next step to run an scenario using LXC virtual machines is downloading an LXC root filesystem, and setting an adequate configuration so that the virtual machines behave as routers. To do so, a filesystem that uses Ubuntu 16.04 as the operating system is downloaded, and moved to the appropriate directory within the VNX filesystem directory, in order to use it in VNX.

3.2.2.1. Filesystem configuration

The filesystem that is used in the LXC virtual machines has been prepared to behave as a router. To do so, the next modifications have been applied:

- Enabling IP forwarding
- Installing and configuring Quagga routing software
- Configuring Quagga

3.2.2.2. Enabling IP Forwarding

Firstly, the Linux Container is started using the *lxc-start* command from the directory where de filesystem is located:

 # lxc-start -n vnx_rootfs_lxc_ubuntu-16.04-v025 -F -f /usr/share/vnx/filesystems/vnx_rootfs_lxc_ubuntu-16.04-v025/config

Once the virtual container has started, the following configurations are done:

To enable the IPv4 forwarding in Linux, the following line is added in the */etc/sysctl.conf* file:

net.ipv4.ip_forward = 1

To save the changes permanently, the following command is executed:

sysctl -p /etc/sysctl.conf

3.2.2.3. Disabling the reverse path filter

The reverse path filter causes problems when packets are forwarded. If the router does not know how to reach the source IP address of a certain packet, the router drops the packet. Since this effect is not desired, and disabling it permanently from the filesystem was not achieved, a configuration tag that disables it from the virtual containers when executed is used.

3.2.3. Installing and Configuring Quagga

The next step consists in downloading and configuring the Quagga routing software suite within the virtual container filesystem.

To do so, an IP address is assigned using the *dhclient* command. This IP address links with a virtual interface in the physical host, that belongs to the same network, and is used as the default route's gateway for the virtual container.

Once the virtual container has Internet access, the Quagga software is downloaded and installed following the instructions from the Quagga support website [5].

The routing protocols that will be used in the project are BGP, IS-IS and RIP. To enable its respective routing daemons, empty configuration files (named as *bgpd.conf* , *isisd.conf,* and *ripd.conf* respectively) are created in directory */etc/quagga* of the virtual container's filesystem. These files must have user and group ownership set to "Quagga". Then, the *daemon* file within the same directory is modified to enable the previous routing daemons as well.

The last step of the Quagga configuration is restarting the Quagga daemon by using the following command, in order to apply the configuration changes:

 # /etc/init.d/quagga restart

At this point, the LXC can be stopped and the next time the LXC starts, the previously named routing daemons will be running.

4. Results

In this section the results are briefly commented. For fully detailed explanations about the behaviour of the scenarios tested, check Appendix A (for BGP) and Appendix B (for IS-IS).

4.1. Testing BGP

The results of the project have been satisfactory, since the BGP protocol, which has been the main focus of the project, has been tested successfully.

The different functions of BGP that has been tested are the following ones:

- Aggregation
- Route Reflectors
- Prefix Lists
- Route redistribution
- Load Balancing
- Attribute manipulation
- BGP Policies

The detailed results that have been achieved, can be found in Appendix A, in the results chapter. This chapter contains the configurations applied to routers, and the results commented, from the questions made in the practices chapter.

4.2. Testing IS-IS

Regarding IS-IS, the results have not been as successful as in the BGP scenarios. Since the IS-IS implementation in Quagga, is in beta version, there are several features that avoid testing the protocol in the scenarios adequately. The main problems encountered in the IS-IS testing, are the following ones:

- Attach Bit not set:

When a Level-1 Level-2 router advertises a Level-1 router with link-state information, it has to set a field in the Link State PDU packet to 1. This field is called Attach Bit. Since topology information is not leaked from Level-2 topology to Level-1 topology in IS-IS, the purpose of the Attach Bit is telling the intra-area routers, that the router setting the Attach Bit to one can be used as a gateway to other IS-IS areas. If this bit is not set, Level-1 only routers do not know where to forward the traffic to a destination outside their own area.

This supposes a very relevant problem, since traffic between Level-1 areas cannot be forwarded from the Level-1 routers. This could be solved manually by setting a default

route, or by announcing the default route from the Level-1 Level-2 routers using another IGP.

- Route redistribution

Another relevant problem is that it is not possible, to redistribute routes to IS-IS. From a router running IS-IS, it is not possible to inject directly connected routes, kernel networks, or routes from other protocols into IS-IS. If IS-IS runs as the IGP in a BGP AS, it will not be able to redistribute the necessary routes so that all routers in the AS can reach the "Next Hop" value and forward traffic to other AS.

For more details, check the scenarios results in Appendix B.

5. Budget

The approximate amount of hours dedicated to the project is 500h. Taking into account that the approximate salary of a junior engineer is about 12,5€/h, this would suppose about 6250€

All the software used in the project is open-source and free. This means that there are no added costs.

The final cost of the project would be **6250€**.

6. Conclusions and Future Developments:

6.1. Conclusions

The project goals have been achieved, since BGP has been tested successfully using LXC virtual machines and VNX virtualization tool. The scenarios loading speed has been improved compared to its preceding tool, VNUML. The behaviour of the BGP protocol implemented in Quagga routing software works as expected, after testing it in different scenarios, and trying the different capabilities it offers. Learning how BGP works has been interesting, as well as useful, since it is a widely used protocol, and currently there is no other EGP that competes against it.

Regarding IS-IS protocol, its implementation in Quagga is far from being complete. At the moment, there are several parts of the protocol that have not been implemented in Quagga. As well as that, bugs have been detected, such as the Attach Bit not being set and packet malformation observed in Wireshark. These issues have impeded developing more scenarios.

Another remarkable aspect, is the improved loading time of VNX's scenarios using LXC, compared to VNUML, that used UML virtual machines.

6.2. Future Developments

Future developments can be added to the current project. If the IS-IS implementation is fully developed, and bugs are fixed, scenarios using BGP and Quagga simultaneously could be implemented, which is what was intended to do in the second scenario in the IS-IS Appendix B. As well as that, the current LXC filesystem can be used to develop scenarios and test other routing protocols implemented in Quagga.

Bibliography:

[1] White, Russ, Danny McPherson, and Sangli Srihari. *Practical BGP*. Boston: Addison-Wesley, 2005. Print.

[2] Halabi, Bassam, and Danny McPherson. *Internet Routing Architectures*. Indianapolis, IN: Cisco, 2000.

[3] VNX Website: http://web.dit.upm.es/vnxwiki/

[4] LXC Website: https://linuxcontainers.org

[5] Quagga Website: http://www.nongnu.org/quagga/

[6] Gredler, Hannes, and Walter Goralski. *The Complete IS-IS Routing Protocol*. London: Springer, 2005. Print.

Appendices:

Appendix A - The Border Gateway Protocol.

Appendix B - Intermediate System to Intermediate System Protocol

Glossary

-AS: Autonomous System

-BGP: Border Gateway Protocol

-EGP: Exterior Gateway Protocol

-IGP: Interior Gateway Protocol

-IS-IS: Intermediate System to Intermediate System

-LXC: Linux Containers

-UML: User Mode Linux

-VNUML: Virtual Networks User Mode Linux

-VNX: Virtual Networks over Linux

Yeray Sainz Lorenzo

Appendix A - Border Gateway Protocol (BGP)

Contents

Chapter 1

Introduction

1.1 Motivation

Routing protocols can be divided in Interior Gateway Protocols (IGPs) and Exterior Gateway Protocols (EGPs). The primary difference between EGPs and IGPs is the place in the network where they provide routing information. That is, within an administrative routing domain (*intradomain*) or between administrative routing domains (*interdomain*). As shown in Figure 1.1, the administrative domain is also called Autonomous System (AS).

Figure 1.1: Autonomous Systems.

One may wonder why to use two different protocols for interior and exterior routing. There are two main reasons. The simple answer is that routing protocols not only provide reachability information, but can also provide policy information. In the context of an autonomous system (AS), generally, policy propagation is not important, since the routers within the routing domain are under a single administrative control and policies can be easily implemented. Thus, an IGP does not need to propagate policy information and this makes it faster and simpler than an EGP. On the other hand, EGPs operate between AS, where it is important to consider routing policies. The **Border Routing Protocol** (BGP) is the standard de facto EGP protocol. Let's examine the network in Figure 1.2 to see how changes in one routing domain may have a negative impact on the operation of another routing domain.

Figure 1.2: Sharing routing information without policy.

In this example, AS1 has decided that the private network prefix 192.168.0.0/24 is one of the destinations for which it shares routing information. So, AS1 distributes this network prefix to AS2 with an IGP. In time, AS2 also partners with AS3 using an IGP. Notice that the routing information provided by AS3 (192.168.0.0/25) conflicts with the internal routing information of AS1 (192.168.0.0/24). There is also another problem, information about AS3 reaches AS1 and vice versa. In general, it is not desirable that information about your internal network leaks beyond your bounds. We might also want to use not the shortest path, but a path that goes across some desired AS. BGP allows to implement these features.

1.2 Main Features

BGP relies on a path vector (known as AS path), that contains a list of autonomous systems a route has traversed while it is announced and reaches new routers and AS. This information is used to avoid routing information loops. If a routers checks that its AS is already is the vector, the route will be discarded, to avoid a loop.

BGP focuses on processing routing information properly. Other features such as error control and transport reliability are delegated to TCP. That is, when two BGP routers exchange BGP related information, the information is exchanged over a TCP connection.

The set of destinations that are announced among two BGP routers (also known as BGP speakers), are called Network Layer Reachability Information (NLRI). These set of destinations, are commonly called prefixes, that are represented by an IP address. A set of prefixes is announced in a single message, and are announced in the same messages if all the prefixes share the same attributes.

BGP may be used to group, for instance, all the routers administrated by a single entity. These could be considered an AS. Within an AS, a priority is the speed of convergence and spreading accurate information about the network topology. However, spreading policy information within an AS is not as important. The reason of this statement is that the policies can be applied manually by the network administrators. These two facts (not needing policy information propagation and convergence speed), make an IGP such as OSPF, a proper routing protocol within an AS. BGP, on the other hand, focuses on policy and network reachability information, as well as scalability. Note that information handled by IGPs (such as topology information) limits scalability.

1.3 Basic Process

The basic design of BGP is fairly simple, which is why it is so flexible. Routes are exchanged among BGP peers by using UPDATE messages. BGP routers receive the UPDATE messages, run some policies or filters over the updates, and then announce the routes to other BGP peers. An implementation is required to keep all BGP updates in a BGP routing table separated from the IP routing table. **In case multiple routes to the same destination exist, BGP does not flood its peers with all those routes; rather, it picks the best route and sends it.** The best route is selected by using the BGP best path algorithm, that is explained at the end of this section, at subsection 1.6.7. In addition to announcing routes from peers, a BGP router can also originate routing updates to advertise internal networks that belong to its own autonomous system. Valid local routes originated in the system, and the best routes learned from BGP peers are then installed in the RIB. The RIB is the last routing decision and is used to populate the FIB.

A simple model of the BGP process involves the following components:

- A pool of routes that the router receives from its peers.
- An Input Policy Engine that can filter the routes or manipulate their attributes.
- A decision process that decides which routes the router itself will use.
- A pool of routes that the router itself uses.
- An Output Policy Engine that can filter the routes or manipulate their attributes.
- A pool of routes that the router advertises to other peers.

1.4 Path Vector

1.4.1 Concept

BGP enables a node to disseminate reachability and policy information about a prefix. Routing information exchanged via BGP supports only the destination-based forwarding paradigm, which assumes that a router forwards a packet based solely on the destination address carried in the IP header of the packet. This, in turn, reflects the set of policy decisions that can be enforced using BGP.

In particular, BGP uses a path vector to disseminate reachability and policy information to its neighbors. At this moment and for the sake of simplicity, we will consider that the reachability and policy information is just the vector of nodes traversed to reach the prefix (network).

The construction of path vectors works as follows. The owner of a prefix announces the prefix to its neighbors and includes its identifier in this announcement. Then, each neighbor receives this information, selects the best path and disseminates again the path vector adding its own identifier. Note that in the middle of the network, a node will probably receive multiple path vectors. With all this information, the node applies its local policy to select one (only one) path vector as the best path. In this case, the best path might be one that is not the shortest one. Also note that the explicit list of traversed nodes serves to avoid loops. If we receive a path vector that already contains our identifier, we just throw away this path.

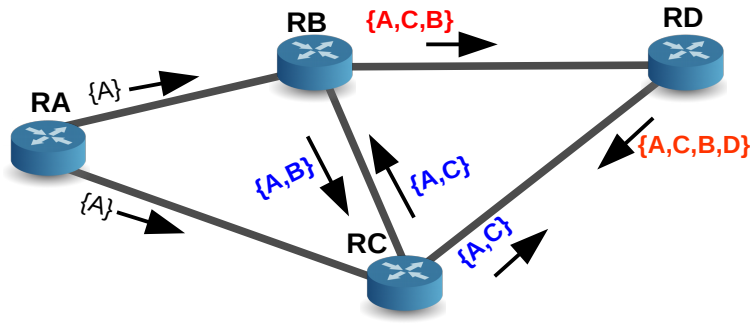


Figure 1.3: Path Vector Concept

Figure 1.3 shows an example of how the announcement of path vectors works.

In this example, router RA sends a path vector with its identifier to its neighbor routers RB and RC. Once these two routers have received the announcements from router RA, routers RB and RC send the path vector to each other, and to other neighbors. As it can be seen, the announcement made from router RB to router RC, contains the path vector {A,B} which are the two routers the announcement has traversed. The same case happens when the announcement is made from router RC to router RB, but with path vector {A,C}. Let's suppose that among the two received path vectors, router RB decides to announce to its neighbor router RD the route to RA through router RC, even though it is longer than the one received directly from RA. On the other hand, router RC decides to announce the shortest path {A,C}. When router RD receives the two path vectors from routers RB and RC, it applies a path selection algorithm and selects path vector {A,C,B,D}. Once it has selected the long path, router RD announces it to router RD,which discards it, since its identifier is already contained within the path vector.

1.4.2 AS path

BGP implements the path vector concept on a larger scale. Rather than treating a single router as a single point in the path to any given destination, BGP treats each autonomous system as a single point on the path to any given destination. As a consequence, the path is going to be a vector of AS. The primary reason BGP treats an entire autonomous system as a single hop in the AS Path is to hide topological details of the AS. An example of a BGP Path Vector is illustrated in Figure1.4.

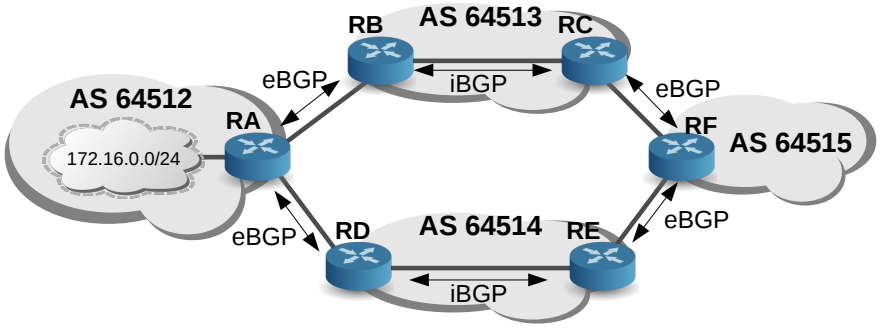


Figure 1.4: Example of a BGP Path Vector

In this example, the network in AS 64512, 172.16.0.0/24, typically referred to as a prefix in BGP, is going to be advertised to other AS. To do so, the routers running BGP, called "BGP speakers", "BGP peers" or "BGP neighbors", have to establish several BGP sessions. A BGP session is also called a "peering session". In this example, there are six existing peering sessions among routers: RA-RB, RA-RD, RB-RC, RD-RE, RC-RF and RE-RF. BGP designers did not reinvent the wheel, and they decided to use TCP between BGP peers when establishing and maintaining peering sessions. Using TCP to transport BGP information allows BGP to delegate error control, reliable transport, sequencing, retransmission, and peer aliveness issues to TCP itself and focus instead on properly processing the routing information exchanged with its peers. Once the BGP session among two peers has been established, routing information between the two peers can be exchanged by using BGP UPDATE messages. Later on with changes in topology, resulting in changes to routing tables, incremental updates are exchanged between the BGP systems. KEEPALIVE messages are sent periodically to ensure the aliveness of the peering sessions. Notification messages are sent in response to errors or special conditions. If a connection encounters an error condition, a notification message indicating the error type is sent and the connection is closed.

There are two types of peering relationships within BGP: interior peering (iBGP), among two routers belonging to the same AS, and exterior peering (eBGP), for peers belonging to different AS. In the example of Figure 1.4, the sessions between RA-RB, RA-RD, RC-RF and RE-RF are eBGP sessions, while sessions between RB-RC and RD-RE are iBGP sessions.

1.5 Peering Sessions

To determine whether a session is an eBGP or an iBGP session, peers compare their AS when establishing the session. If their respective AS are the same, an iBGP session will be established, otherwise it will be an eBGP session. There are important differences between iBGP and eBGP peering sessions, mainly due to the fact that BGP treats an AS as a single entity in the path vector. This implies that BGP cannot detect loops inside the AS. If not configured properly, external sessions can also form loops. As a result, BGP has strong rules to prevent internal and external loops.

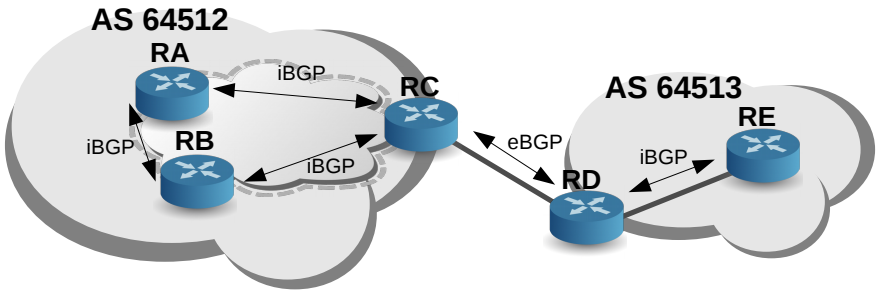


Figure 1.5: Types of peering sessions among BGP routers.

As shown in Figure 1.5, routers belonging to the same AS, establish iBGP peering sessions among them. For instance routers {RA,RB,RC} and {RD,RE} establish iBGP sessions among each other. Routers RC and RD establish an eBGP sessions, since each of these two routers belongs to a different AS.

1.5.1 Establishing a Peering Session

When a peering session is going to be established among two BGP routers, a negotiation process takes place and a set of messages are exchanged. There are 4 kinds of messages:

- OPEN
- UPDATE
- NOTIFICATION
- KEEPALIVE

Once a TCP connection has been established, a BGP peer sends an OPEN message and waits for a response with another OPEN message from the router it is trying to establish a BGP session with. Once the OPEN response message is received, if there are no conflicts (e.g. unacceptable AS number or bad version), a KEEPALIVE message is sent to the neighbor, and the neighbor should reply with another KEEPALIVE message in order to successfully establish the BGP peering session and end the negotiation process. KEEPALIVE messages are then sent periodically at the rate set by the KEEPALIVE timer. Once the BGP session has been established, UPDATE messages are exchanged among the neighbors UPDATE messages announce the best known BGP routes. Once the routes are announced, KEEPALIVE messages exchange resumes between peers, in order to maintain the peering session alive. If any kind of unexpected event or error happened during the negotiation process,are or once the session has been established, NOTIFICATION messages are used to indicate the kind of problem that has occurred, and the BGP session is then closed.

1.5.2 iBGP sessions

- **Definition:** BGP sessions between peers within a single autonomous system are referred to as interior BGP, or iBGP, sessions.

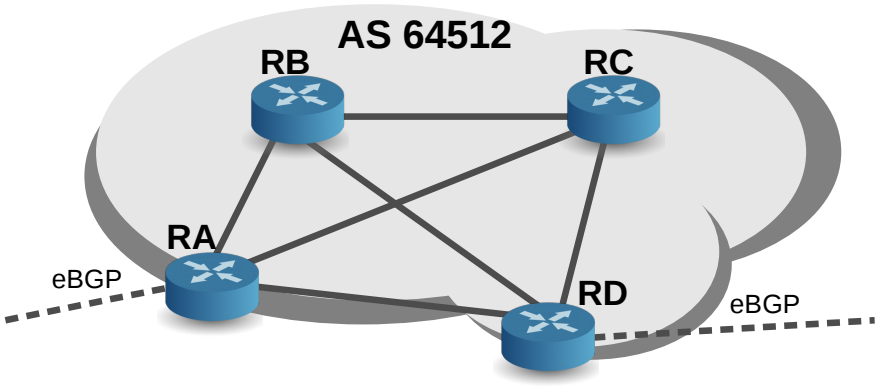


Figure 1.6: Physical iBGP full mesh.

- **Connectivity:** The simplest configuration, which is loop-free and easy to deploy, is to have a full mesh of physically connected iBGP routers within the AS, as shown in Figure 1.6. In this scenario, there is an iBGP session established between each pair or routers, through each physical link.

  However, the physical full mesh model cannot be deployed easily in a growing network, it is not easily scalable. An AS with different BGP routers can be configured in a way that BGP peering sessions are done in a non-physical full mesh scenario, as shown in Figure 1.7.

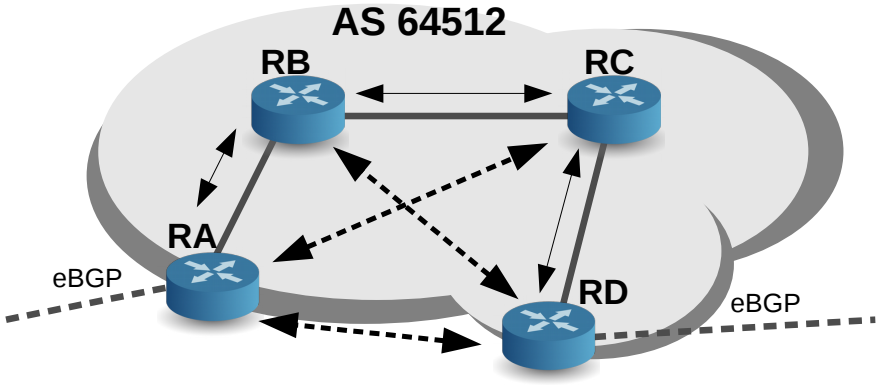


Figure 1.7: Non-physical iBGP full mesh.

  In this scenario, the iBGP peering sessions are indicated with the arrows. The existing iBGP sessions among routers that are not physically connected are indicated with the dashed lines. In this simple case, if routers RB and RC know how to reach RD and RA respectively, the iBGP sessions can be established without any problem. However, in an AS where an IGP is running, and the sessions are established through non-BGP routers the configuration might be more tricky, and loops may occur.

- **iBGP Protocol:**

  - Routes learned from an iBGP peer are not advertised to other iBGP peers. This prevents routing loops within the autonomous system (but full mesh is then necessary).
  - In general, the attributes of paths learned from iBGP peers are not changed. The best path chosen by the routers of the AS **must be consistent (the same)** to prevent routing loops inside the AS.
  - The AS Path is not manipulated when advertising a route to an iBGP peer; the local AS is added to the AS Path only when advertising a route to an eBGP peer.

- The BGP next hop is normally not changed when advertising a route to an iBGP peer. However, it can be changed, for example, to forward the inbound traffic through a certain router.

1.5.3 eBGP sessions

- **Definition.** BGP sessions running between peers in different autonomous systems are referred to as exterior BGP, or eBGP sessions.
- **Connectivity.** By default, eBGP peers have to be physically connected, i.e. adjacent to one another. An eBGP peer will drop any BGP update message from its external BGP peer if the peer is not physically connected.
- **eBGP Protocol.**
  - UPDATE messages received by an eBGP peer are not advertised to another peer within the AS, if the AS number already exists in the AS path.
  - BGP attributes can be modified by applying policies before advertising the routes.
  - The next hop in the UPDATE message is changed to the local peer termination IP address, when advertised to another eBGP peer in a remote AS.

1.5.4 Peer identifier

The identifier of a peer or ROUTER_ID is the IP address that the peer uses in its BGP messages. In particular, a peer many times has to set the NEXT_HOP attribute to its ROUTER_ID. The ROUTER_ID could be the address of any of the routers' interfaces. Keep in mind that the stability of the neighbor connection depends on the stability of the IP address you choose. If you choose the IP address that belongs to an Ethernet card that has some hardware problems and is shutting down every few minutes, the neighbor connection and the stability of the routing system will suffer. Most implementations provide the capability to configure a virtual interface, referred to as a loopback interface, that is supposed to be up at all times. Tying the BGP neighbor connection to a loopback interface will ensure that the BGP session is not reliant on any hardware interface that might be problematic.

Using loopback interfaces is not necessary in every situation, for example, if external BGP neighbors are directly connected and the IP addresses of the directly connected segment are used for the neighbor negotiation, a loopback address is of no added value because if the physical link between the two peers is problematic, the session will break with or without loopback. The default ROUTER_ID for a router is vendor-specific, but for most implementations, the loopback address if one is configured is selected as default; otherwise, it's the highest IP address on the router.

1.5.5 Prefix Announcement Example

In Figure 1.8 a prefix announcement example is shown. Router RA in AS 64512 generates two UPDATE messages including the prefix, the next hop and the AS path for its eBGP peers in AS 64513 and AS 64514:

```
(1) UPDATE{prefix:172.16.0.0/24,next hop:10.0.0.1,AS path:64512}.
(2) UPDATE{prefix:172.16.0.0/24,next hop:10.0.1.1,AS path:64512}.
```

In our example, routers RB and RD are directly connected with their respective peers RC and RE by means of some data link layer. They have also established iBGP sessions with these iBGP peers and using these sessions they send the following UPDATE messages:

```
(3) UPDATE{prefix:172.16.0.0/24,next hop:10.0.0.1,AS path:64512}.
(4) UPDATE{prefix:172.16.0.0/24,next hop:10.0.1.1,AS path:64512}.
```

As you observe, UPDATE messages are not altered in iBGP sessions. Remember also that iBGP peers do not re-advertise UPDATE messages. Then, peers RC and RE acting as eBGP peers of RF send the following messages:

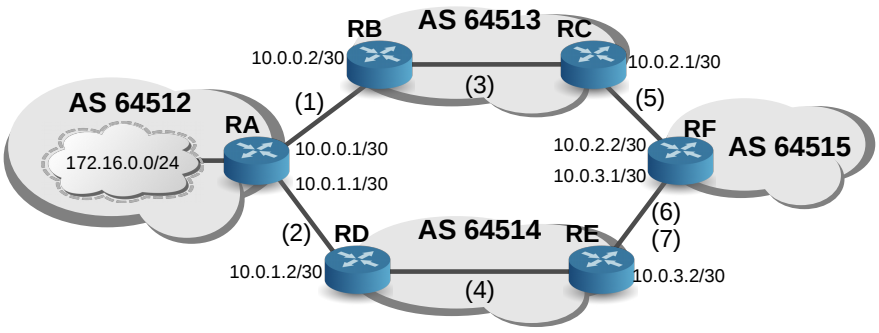


Figure 1.8: Example of a BGP Path Vector continued

```
(5) UPDATE{prefix:172.16.0.0/24,next hop:10.0.2.1,AS path:64512,64513}.
(6) UPDATE{prefix:172.16.0.0/24,next hop:10.0.3.2,AS path:64512,64514}.
```

As you observe, in the eBGP session, the router changes the next hop field and also adds its AS to the AS path. Then, the peer RF applies a best path selection algorithm to choose one of the two paths received as the preferred. Let's assume that after applying the algorithm to the two routes, the preferred one is the announced by router RC. Since BGP routers just announce the best considered route, the only outbound UPDATE message from router RF will be the following one:

```
(7) UPDATE{prefix:172.16.0.0/24,next hop:10.0.3.1,AS path:64512,64513,64515}.
```

When this last UPDATE reaches router RE, it will again apply the best path decision algorithm and decided which is the best one among the two received. Since there are less AS in the AS path, let's consider that router RE selects as the best route the one received firstly from its peer router RD. That would mean that the router received from router RF would not be announced.

Note that for policy reasons, a peer might prefer a route containing a longer AS path, and BGP provides the possibility of selecting the desired path among all the possible. The mechanism to select the best path among several received, is done by the BGP best path decision algorithm, detailed in section 1.6.7, and is done in the locally. Also notice that as we are considering a path in which there are only AS numbers, a peer cannot tell what the path through a particular AS looks like, only that the destination is reachable through the AS. On one hand, this is a way of achieving privacy. On the other hand, it is also a way to achieve more flexibility to perform AS internal changes without externally affecting the routing service.

1.6 BGP Attributes and Capabilities

In IGPs, we are used to decide the route with a single attribute. On the other hand, BGP is very flexible and has a relatively big set of attributes to decide the "best path". These attributes fall into two broad categories: **well known** and **optional**. Well known must be implemented by any BGP router. Well known attributes can be further divided into **mandatory** and **discretionary**. Mandatory attributes must be included in every single BGP update, while the administrator of each router can decide to include or not a number of discretionary attributes in a BGP update.

- Examples of well known mandatory attributes are: ORIGIN, AS-PATH and NEXT-HOP.
- Examples of well known discretionary attributes are: LOCAL-PREFERENCE and ATOMIC AGGREGATE.

Regarding optional attributes, these might not be implemented in a certain BGP router. The implementor can decide to implement some optional attributes or even create its own (proprietary) optional attributes. Optional attributes can be further divided into **transitive** and **non-transitive**. The **transitive** attributes, whether those attributes are recognized or not, have to be advertised to peers. In the case of receiving a BGP update that contains one or more optional **non transitive** attribute, the UPDATE message should be advertised without the unrecognized attributes.

- Examples of optional transitive attributes are: COMMUNITIES and AGGREGATOR
- Examples of optional non transitive attributes are: MED and CLUSTER LIST

In the following subsections, the most relevant attributes are explained.

1.6.1 Origin Code

This well-known mandatory attribute indicates the way in which the prefix was injected to BGP. It can indicate IGP, which means the prefix has been learned from an interior gateway protocol, which is the most common case,EGP, which means that the prefix was learned from the obsolete EGP protocol that BGP replaced,or INCOMPLETE, if the origin of the prefix is unknown. INCOMPLETE value of the origin code is usually set when the prefix is learn, for instance, via aggregation or redistribution.

1.6.2 AS Path

As previously explained, the AS_PATH well-known mandatory attribute lists all the autonomous systems a prefix or group of prefixes in an UPDATE messages have passed through. The AS is added to the AS path at the edge router/s of an autonomous system, that is, when the edge router/s send an UPDATE to a neighbor eBGP peer.

1.6.3 Next Hop

NEXT_HOP is a well-known mandatory attribute that is set at the edge router of an autonomous system, when a prefix is advertised to a neighbor eBGP peer located in a remote AS.

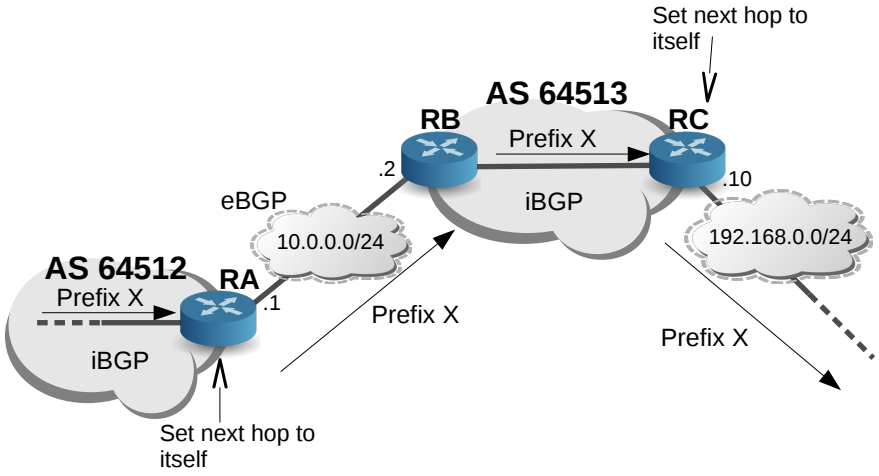


Figure 1.9: BGP next hop.

Figure 1.9shows an example of BGP next hop behaviour.

When a certain prefix is announced to a neighbor AS, the value of the BGP next hop attribute is set to the IP address value of the edge router it is leaving. For instance, in the example above, when prefix X is announced via an eBGP session from AS 64512 to AS 64513, the value of the next hop attribute will be set to 10.0.0.1. This IP address, is the value of the edge router RA's interface through which it is advertised to the neighbor autonomous system. When router RB receives the UPDATE message containing this prefix,it will not modify the next hop attribute value, since it will advertise it to a router within the same AS 64513, that is, to router RC. Once again, when router RC advertises the prefix to a neighbor autonomous system, will set the next hop value to 192.168.0.10. Note that routers within an autonomous system must know how to reach the next hop address. For instance, if router RC wants to reach the next hop, when forwarding traffic to prefix X, it must firstly have the network 10.0.0.0/24 in its routing table. This could be achieved, for instance, by redistributing the connected routes in router **RB** into an IGP running within AS 64513.

Next hop value can be also used to forward the inbound traffic to an AS through a particular edge router, as shown in Figure 1.10

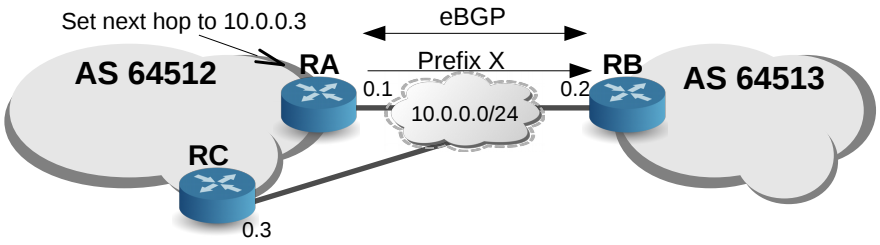


Figure 1.10: Setting different next hop example.

In Figure 1.10, there is an existing eBGP peering session among routers RA and RB. If the next hop value in router RA is set as the other edge router RC interface's address, for a certain range of IP addresses (let's assume it is Prefix X in Figure1.10) the inbound traffic flow will have a particular behaviour. All the traffic forwarded to prefix X, will flow through router RC, instead of flowing through router RA. It might be useful, for instance, to reduce the CPU load in router RA.

1.6.4 Multi Exit Discriminator

Multi Exit Discriminator (MED) is an optional non-transitive attribute, used to hint neighbor autonomous systems about which is the preferred entry point to a certain AS. For instance, an autonomous system will compare the MED values received from different edge routers belonging to the same autonomous system, and use the lowest MED value as the preferred route (if there is no other BGP metric set, with a higher preference in the BGP decision algorithm, such as the local preference).

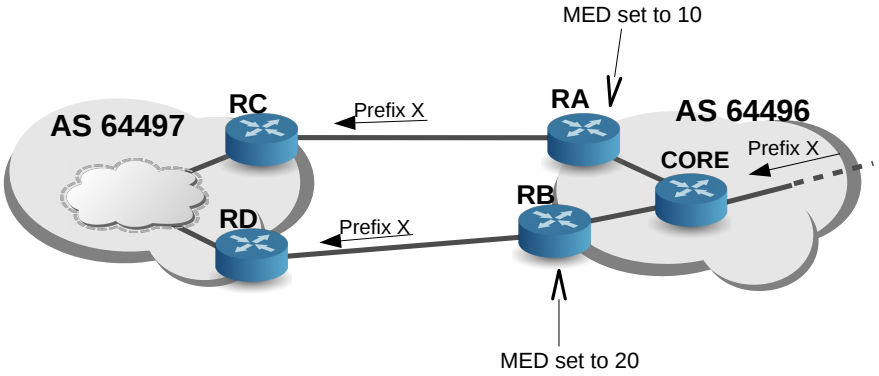


Figure 1.11: Multi Exit Discriminator example.

Figure 1.11 shows an example in which a certain prefix X, is received at CORE router of AS 64496, and it is advertised to the AS's edge routers, RA and RB via iBGP. These two edge routers, are going to advertise this prefix to the edge routers of the neighbor AS 64513, via eBGP. The MED value in the edge routers of AS 64512 has been set to 10 and 20 in RA and RB respectively. This means that, BGP routers in the neighbor AS 64513 will know that the preferred exit point within AS 64513 for AS 64512 will be through router RC, since the lowest MED value has been announced from router RA.

This will make routers within AS 64513 prefer forwarding packets to prefix X (or a set of prefixes, depending on the configuration) through router RC, if there is not a higher preference attribute (such as weight or local preference), that is decisive when applying the BGP best path decision algorithm.

1.6.5 Local Preference

This well-known attribute is used within an AS, to determine which is the preferred path for a certain prefix or prefixes. While the MED attribute was a hint of the preferable route for a neighbor autonomous system, the local preference (LOCAL_PREF) is configured within an autonomous system to be used within itself. Furthermore, this attribute has a higher preference when computing the best path decision algorithm. The default value for this attribute is 100, and the higher value it has, the most preferable a certain route is.

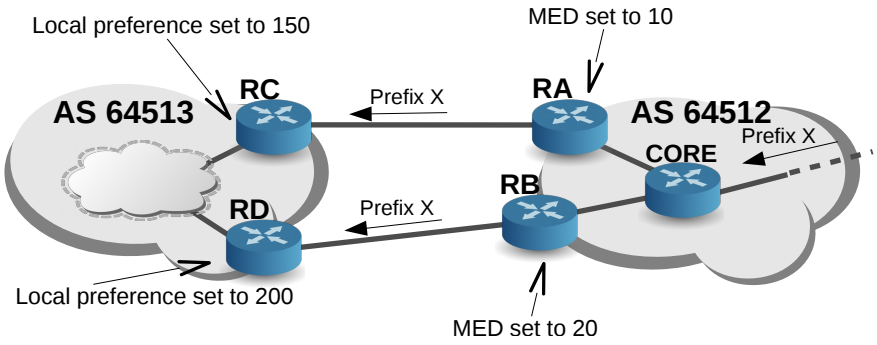


Figure 1.12: Local preference example.

In the example in Figure 1.12, prefix X is received from two edge routers RA and RB in AS 64512. These two routers are also advertising two MED values, as in the example in Figure 1.11. Meanwhile, in AS 64513, routers RC and RD set local preference values, 150 and 200 respectively, for the prefixes received from its eBGP peers, routers RA and RB. This will make packets forwarded to an address within the prefix X range through router RD, instead of through router RC. The reason of this behaviour is that, when applying the BGP best path selection algorithm, the local preference value has a higher preference than the MED values, so MED values are ignored in this case.

Note that, in the local preference case, the higher the value is, the most preferable a certain route is (the opposite case of MED, that the lower the value is, the most preferable a route is).

1.6.6 BGP Aggregation

When a BGP router has to advertise, for instance, two routes such as 10.0.0.0/24 and 10.0.1.0/24, these two routes can be advertised as a single route 10.0.0.0/23. As a consequence, the attributes in the aggregated route announcement may be different.

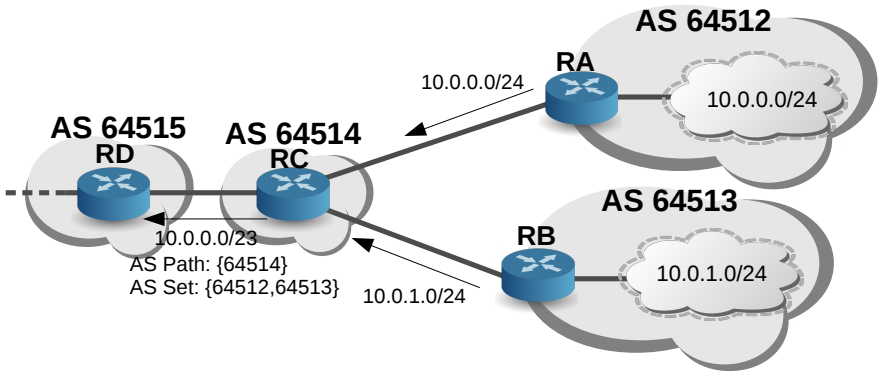


Figure 1.13: Aggregation example.

In Figure 1.13, routes 10.0.0.0/24 and 10.0.1.0/24 are aggregated into 10.0.0.0/23 in router RC. Once these two routes are aggregated, and announced to router RD to router RC, note that the AS path attribute cannot be set. If

the route were announced separately, the AS path for 10.0.0.0/24 would be {64512,64514} while the AS path for 10.0.1.0/24 would be {64513,64514}. On one hand, the AS path attribute, indicates the AS a certain prefix has traversed, in a sequential order. This means that setting one of the two previous AS paths, or an AS path containing all the AS both routes have traversed, would be incorrect. On the other hand, there is an attribute that indicates the AS path a route traverses, but this time, not in a particular order. To solve this problem, a combination of both AS path and AS set attributes can be applied. When the aggregated route is announced, the solution would be including the AS from the non-aggregated routes, in the AS set attribute. From that point, the AS path attribute would contain the AS that the aggregate route traverses, that is, AS 64514.

1.6.7 BGP Best Path Decision Algorithm

When a BGP router receives an announcement from a peer, it can happen that the same prefix is also received from other peers. When this situation happens, the BGP router must apply a certain algorithm to select which is the best path to a certain network among the several received. This algorithm is based on comparing a series of attributes in order to select a route. Each attribute used in the BGP Best Path Decision Algorithm, has a certain preference among others, so that when there is a "tie" among a certain attribute, the next attribute in the preference list is compared.

Notice that the BGP best path decision algorithm takes place locally in a router. That is, when there are two equal prefix in the BGP table (not the routing table), this algorithm is used to select the best route. Once the best route is selected, it is announced to its peers.

1. **Weight check**

   The weight attribute is set within a router and is not announced to any of its peers. For instance, a router can be configured to set a higher weight for a pool of routes received from a particular peer, so that traffic to this pool of routes is always forwarded through that particular peer, regardless of the other attribute's influence.

2. **Local preference check**

   Select the route with the highest local preference value.

3. **Local route check**

   Networks announced via the *network*, *redistribute* BGP commands, or IGP redistribution are preferred over routes learned from a BGP peer.

4. **AS path length**

   Prefer the shortest AS path among the different routes received.

5. **Origin check**

   Prefer the IGP originated routes over the EGP originated routes, and prefer the EGP routes over the routes defined as INCOMPLETE.

6. **MED check**

   Select the route with the lowest MED value.

7. **eBGP over iBGP paths**

   Paths received from an eBGP peer are preferred.

8. **Lowest IGP metric to BGP next hop**

   The path that has the lowest metric to the BGP next hop, is preferred.

9. **BGP Multipath**

10. **Oldest announcement first**

    Prefer the path that was received first.

11. **Lowest neighbor router ID**

    Prefer the path that comes from the BGP router, with the lower router ID.

12. **Minimum cluster list**

    If the router ID is the same, preferred the route with the lowest cluster list length. Note that this is only applicable in a route reflectors environment.

13. **Lowest neighbor address**

    The route announced by the neighbor that has configured the lowest IP address on its BGP configuration is preferred.

Chapter 2

BGP Routing Design

In this chapter it is explained how to solve routing problems and how to design routing scenarios. The topics explained are about attribute manipulation techniques and route filtering usage to influence the BGP decision process, and design criteria (redundancy, load balancing and symmetry) when developing routing policies and integration between IGP and BGP. In addition, some troubleshooting is also presented.

2.1 Routing at the AS's Edge

In this section it is discussed how an AS can be connected to another AS, and different possible solutions to deal with redundancy, symmetry and load balancing issues. These cases can be applied when, for instance, a certain costumer wants to connect to the internet via a service provider by one or multiple physical connections. Firstly, techniques to make the inbound traffic to an AS go through a link or another are explained, then, how to balance the inbound traffic among different links. It is also explained how to control the outbound traffic flow from a certain AS, by using the IGP running in the AS, or by using BGP.

2.1.1 Inbound Traffic Flow Control

The inbound traffic flow is all the traffic that enters in an AS from others. The administrators of an AS might want to manage the inbound traffic if the AS is, for instance, connected to the internet via two different links to its service provider. The AS administrator might want to receive inbound traffic to its AS from one or another link, but the inbound traffic is forwarded by routers in another AS out of the administrator's control.

A common misconception about IP traffic flows is considering that are symmetric,that is, for instance, that the reply messages of a certain request are sent back through the same path as the requests. This means that inbound and outbound traffic flows must be controlled and managed separately, since those traffics are not symmetric.

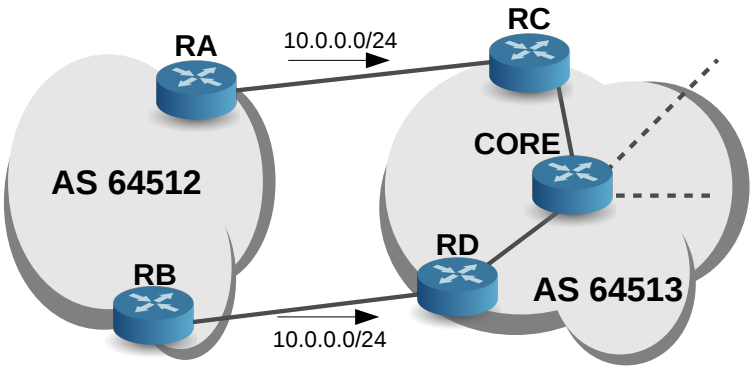


Figure 2.1: Dual homed AS.

In Figure 2.1, devices in the costumer AS 64512 get internet access through routers RA and RB edge routers. These two routers, at the same time, peer with service provider's routers RC and RD, that belong to service provider's AS 64513. In the case the same prefix is announced by both routers RA and RB, the CORE router in the service provider's AS will only select one of the two routes to reach the prefix network (since BGP always chooses a single path to a destination). This will make all traffic forwarded to the prefix network (10.0.0.0/24) that has been announced, flow through one of the two links, which might not be the desired behaviour.

In the following subsections, different possibilities to control the inbound traffic flow are going to be explained.

2.1.2 Inbound Traffic Load Balancing

Different methods are going to be described to achieve load balancing among the multiple paths that can exist between two different Autonomous Systems. These methods include both BGP attribute manipulation and prefix length modifications.

Multi Exit Discriminator (MED)

One way to control the inbound traffic to an AS we are managing, is using the **Multi Exit Discriminator** attribute when the prefix is announced to the provider's AS edge routers. The MED can be used, in this case, to hint the neighboring AS which is our preferred inbound path for a particular prefix. Nevertheless, it might occur that the neighboring AS resets the MED value at its edge routers, which means that the MED we are announcing would have no effect in the neighboring AS's forwarding. In the case the value is not reseted by the neighbor edge routers, and the value is used by the AS that receives the MED values, it should be propagated from the edge routers to its iBGP peers.

Communities

Another way to control the inbound traffic flow to an AS, is using communities. For instance, if the prefix announced to the neighbor AS is sent together with a certain community number at one of our AS's edge routers, the neighbor AS's edge router may interpret it as setting the number as its local preference. The details on what communities can be used to do so and its semantics should be discussed with the neighbor's AS administrators.

Prepending extra hops

A third option to control the inbound traffic is prepending extra hops onto the AS path to make a certain path preferable. This is as simple as adding extra AS numbers in the AS path, which makes the AS path longer, and therefore, less preferable for the neighbor AS to forward the traffic to one of the two links. The prepended AS number in the AS path, is commonly the local AS number of the router that makes this operation.

Using Longer Prefix Advertisements

Another option that might work to achieve load balancing in both links, is announcing longer prefixes.

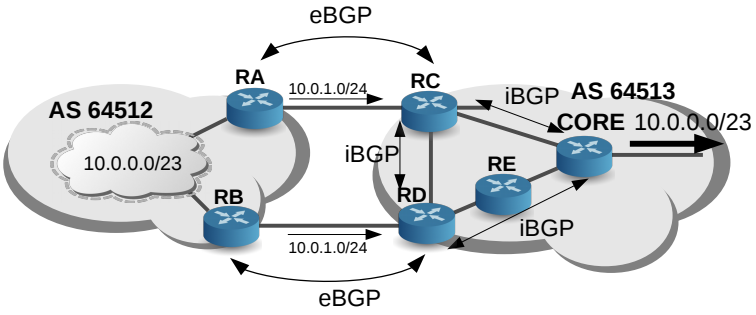


Figure 2.2: Dual homed AS.

In Figure 2.2, an example of applying longer prefix advertisements is shown. Let's suppose that the AS 64512, which we are administrating, has been assigned the 10.0.0.0/23 pool by our service provider. If this prefix is announced, the inbound traffic to the AS we are administrating would depend on the provider's AS configuration. Since the metric provided by the IGP in the service provider's AS, or perhaps the weight parameter at the CORE router may make going through router RA more preferable, the CORE router would choose forwarding packets to 10.0.0.0/23 through routers RC and RA (Note that router E is non-BGP). A technique to try balancing the inbound traffic could be "splitting" this pool into 10.0.0.0/24 and 10.0.1.0/24, and then announce each prefix in a different link to AS 64513, which is our service provider's AS.

When the prefixes are announced in this way, the CORE router at the provider's AS, its routing table will contain two different paths to two different prefixes. If the hosts in our network are equally distributed among the two prefix ranges, and each host receives about the same volume of traffic, the inbound traffic in each link should be about equal.

2.1.3 Outbound Traffic Flow Control

In the next lines, methods to control the outbound traffic of an AS we are administrating are described. In the inbound traffic flow case, the traffic control is strongly tied with the peers' BGP and route aggregation policies. In the case of the **outbound traffic flow**, it is far more easy to control the traffic. This is due to the fact that IP routing protocols are designed to control the adjacent next hop toward any given destination, and controlling the next hop taken toward any given destination also controls the path the traffic takes.

Outbound Traffic Flow Control Using an IGP

In an scenario where two edge routers are announcing the default route inside an AS via an IGP, the different routers in this AS will forward packets that its destination address is not contained in its routing table, to one of these two routers that is announcing the default route, using the criteria of the metric imposed by the IGP. By modifying the metrics and the locations from which the default route is being advertised, the amount of traffic that is transmitted outbound on each link, can be controlled.

Another option to improve the outbound traffic flow control, could be the one shown in the following scenario:

In the scenario in Figure 2.3, by making longer prefix advertisement in each of the two edge routers, the outbound traffic flow can be controlled as well. To do so, for instance, an edge router could advertise the default route plus 0.0.0.0/2 and 64.0.0.0/2 networks, while the other edge router may advertise 128.0.0.0/2 and 192.0.0.0/2 plus the default route.

By configuring these static routes, the packets to be forwarded will match a more specific route, and will be forwarded through one of the two outbound links, instead of forwarding through the shortest-metric edge router at all times. Note that the static route distribution shown as an example may not be optimal, since the outbound traffic of a certain AS will probably not be distributed equally in the whole IP addresses range.

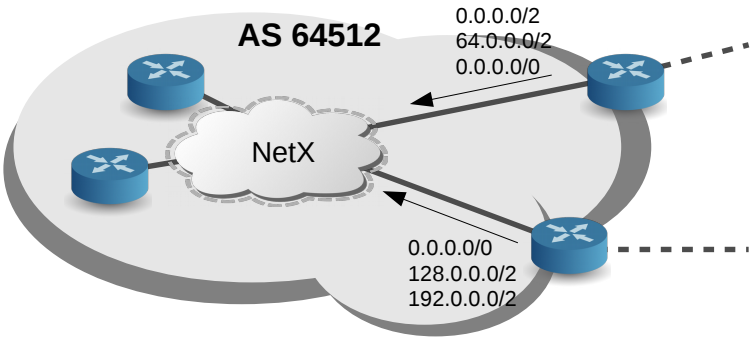


Figure 2.3: Controlling the outbound traffic announcing more precise routes.

Using BGP to Control Outbound Traffic Flow

As happens with traffic control using IGPs, there are several ways to control outbound traffic flow using BGP. Since BGP Confederations and Route Reflectors have not been presented yet, simpler solutions can be implemented.

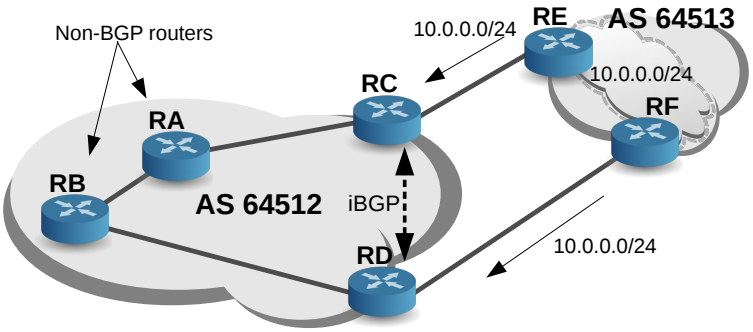


Figure 2.4: Outbound load sharing using BGP.

In the scenario in Figure 2.4, let's suppose that we want to control the outbound traffic flow in AS 64512. The interior non-BGP routers in AS 64512, that is, RA and RB, receive the default route from routers RC and RD via an IGP. This will make routers RA and RB forward all the traffic through routers RC and RD respectively. Note that there is an IGP running within AS 64512, so establishing iBGP sessions among routers RC and is not a problem.

Now, let's suppose that router RF and RE announce the prefix 10.0.0.0/24, which is later advertised by router RC to router RD, and vice versa, via iBGP. Let's suppose that for some BGP policy reasons, router RC selects the route to 10.0.0.0/24 through router RD, instead of through router RE (which would be the common behaviour).

If router RA forwards a packet to 10.0.0.0/24, it will do it through router RC, since it has received the default (and only) route from router RC using an IGP. Once router RC receives the packet, it will see that it has to forward the packet through the other edge router in the AS (router RD), because of the policies it has applied.

The packet then, will return back to router RA: There is a forwarding loop.

Different solutions can be applied to solve this issue:

- Running BGP in routers RA and RB, and establishing full-mesh iBGP peering sessions within AS 64512. Note that physical links among each pair of routers in AS 64512 would not be necessary, since there is a IGP running in the AS. This would be the recommended solution.
- Setting a physical link between routers RC and RD would make RC forward the packet directly to router RD, not causing any forwarding loop.
- Injecting the BGP routes into the IGP, and setting BGP synchronization in the edge routers RC and RD. Injecting the BGP routes into the IGP would make routers RA and RB know a more specific route than the default one,

but there is still a problem. Even though routers RA and RB have a more specific route, their nearest gateways are RC and RD respectively. That means that the forwarding loop within routers RA and RC will still exist. At this point, it is necessary to set BGP synchronization in AS 64512 routers. What synchronization does, is not announcing routes through iBGP sessions belonging to the same AS. Now, router RC will not know the route through router RD, and will forward the packet through router RE. Note that in the original case, router RC decided among two possible routes in the BGP table, and after applying BGP policies, it chose the route through router RD. After applying synchronization, router RC just has a single route to 10.0.0.0/24 in its BGP table (through router RE), so it will choose the only one available.

BGP synchronization might be useful to fix forwarding loops within an AS containing non-BGP routers after injecting BGP routes into the IGP. However, it is not recommended; injecting many BGP routes into the IGP, may lead to resources exhaustion.

2.2 iBGP Core Design

In this section it is described how to design and implement an iBGP network core. Up to this chapter, BGP peerings inside an AS, that is, iBGP peerings, have been fully meshed. The reason to make all the BGP routers inside an AS having an iBGP peering session to other BGP routers, is that an iBGP peer that receives an UPDATE message from another iBGP peer, only transmits it to its eBGP peers. If there is not a full mesh, it may happen that some network prefixes are not advertised through other AS that should receive this information. If the iBGP peers are not in a full mesh, routing information loops may occur, caused by the duplication of routes, that may also cause forwarding loops.

To "relax" the iBGP full mesh restrictions, **Route Reflectors** can be used , as well as **BGP Confederations** to group separate routing domains, that can belong to different AS so that the group of AS is seen as a single AS from outside of the BGP Confederation.

2.2.1 Route Reflectors

The utility of **route reflectors** is, reorganizing an iBGP system, not having the necessity to have a full mesh iBGP sessions among all iBGP speakers, but building a hub and spoke and relaxing the route advertisement rules, that is, now advertisements between two iBGP speakers may not exist.

Route Reflection Rules

A route reflector has two type of peers: client and non-client peers.

- Client peers: These BGP routers don't need to be fully-meshed with all the BGP routers in the AS they belong to. A client peer, establishes iBGP peering sessions just with the route reflectors it is client of.
- Non-client peers: These BGP routers establish iBGP peering sessions with the other non-client peers and the route reflectors in the AS.

One or a set route reflectors and its client peers form what it is called a route reflection cluster. The reflected routes are advertised after running the best path selection algorithm in the router, and its rules are as follows:

- If the route was received from a non-client peer, reflect the route to all client peers.
- If the route was received from a client peer, reflect the route to all non-client and client peers.

The common default behaviour is not maintaining iBGP sessions between clients in the same reflection cluster.

In Figure 2.5, a simple scenario containing route reflectors within an AS is shown. Router RD is a route reflector, while routers RE and RF are the route reflector's clients.

Let's analyze what would happen across the AS if an advertisement originating at router RA was generated:

- Firstly, the announcement is received by router RB, and sent to its iBGP peers, that is, routers RC and RD.

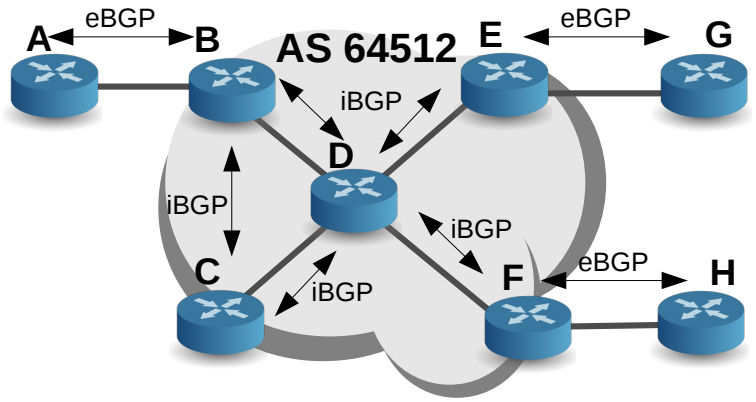


Figure 2.5: Simple scenario containing route reflectors.

- Router RC will not advertise it, since there are no eBGP peering sessions involving this router.
- Router RD has received an advertisement from a non-client peer (router RB), so it is going to announce the advertisement to its client peers (routers RE and RF as the route reflection rules establish.
- Routers RE and RF make the announcement to their respective eBGP peers RG and RH.

Now, the case when the advertisement is originated at router RG can be analyzed:

- Router RE receives the announcement from router RG, and announces it to its iBGP peers, that is, to router RD.
- Router RD has received the announcement from router RE, which is its client. Following the route reflection rules, when an announcement is received from a client peer, it has to be announced to all client and non-client peers, so it will be advertised to routers RF, RB and RC.
- Router RF will transmit it to its eBGP peer, router RH.
- Router RC will not transmit it to any peer, since it does not have any active eBGP peering session.
- Router RB will make the announcement to its eBGP peer, router RA.

In the case that the same route was originated in both routers RG and RH, both announcements may reach router RD. Then, router RD would apply its best path selection algorithm and select, for instance, the announcement received from router RE. This announcement would be sent to its non-client peers, that is, routers RB and RC, which would proceed in the same way as in the previous example. When router RF would receive the announcement, it would discard it since it would prefer the route learned through its eBGP peer (BGP best path selection algorithm, step 6).

Attributes Added to Prevent Loops

In a large AS where route reflectors are employed, routing information loops can appear. In order to prevent these loops within an AS where there are route reflectors, the**Originator ID** and **Cluster List** are used.

The Originator ID contains an identifier from the router that announced a prefix to a route reflector, which is usually the same value as the router ID, but can be configured to another explicit value. This value is set at the route reflector, before the prefix is announced to other client and/or non-client peers.

The Cluster List is a set of authenticators that contain the Cluster IDs of all the route reflectors a prefix has gone through within an AS. These Cluster ID values that are contained in the Cluster List, are usually obtained (as well as with the Originator ID) from the Router ID parameter from the route reflectors.

Route Reflectors Deployment

If we want to build an scalable iBGP core, we should include route reflectors to reduce the number of iBGP sessions. Other aspects such as redundancy can be treated when using route reflectors, that is, for instance, the core should not rely in a single route reflector to reflect the prefixes to other peers in the core. Having more than one iBGP peering session with several route reflectors will make the core more redundant and reliable.

Another factor that has to be considered when deploying multiple route reflectors within an AS, is considering if sets of multiple route reflectors should be set within the same **Cluster ID** or not.

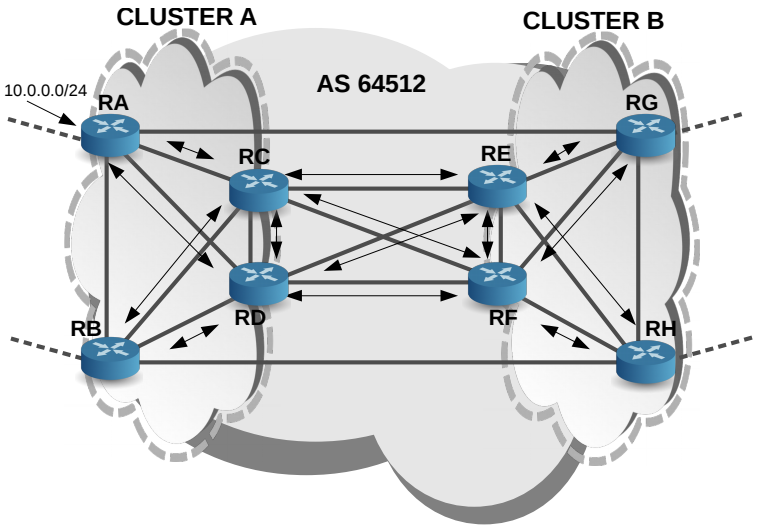


Figure 2.6: Configuration containing route reflectors and reflection clusters.

In the example in Figure 2.6, the are two different reflection clusters (A and B), four route reflectors (RC, RD, RE, RF) and its clients. routers RA and RB are both clients of route reflectors RC and RD, while router RG and RH are clients of route reflectors RE and RF. Arrows indicate iBGP peering sessions.

- **Redundancy**:

  The reflections in this system could work with just a single route reflector in each cluster, but it might present some issues if it fails. In the case the single route reflector may fail, for instance, in the cluster A, routes could not be reflected between clusters, neither within clients in cluster A (note that there is no iBGP peering session among clients of a route reflector, as an ordinary configuration). Routes reflected within cluster B would still work.

  A solution to this problem is the one shown in Figure 2.6: By setting a second route reflector in each cluster, reflected routes can still be announced to route reflector's clients even if one fails, as well as to the neighbor reflection cluster.

  Setting other redundant links within route reflectors makes the scenario more reliable in the case one or more link fail.

- **Cluster IDs:** The reason to configure each pair of route reflectors belonging to the same cluster, is to avoid more than one copy of a prefix in each route reflector client. If, for instance, within a reflection cluster, where each route reflector does not belong to the same cluster, an announcement is reflected by each of the two route reflectors to one of its clients, the client will store the **two** announcements, which would mean storing duplicate information (and may be critical in the case of thousands of prefixes announced). On the other hand, if both route reflectors are set in the same cluster (same cluster ID value), clients will just have one prefix installed, since the announcements received belong to the same cluster.

If the prefix 10.0.0.0/24 is announced from router RA, the announcements made within AS 64512 would be the following ones:

- Router RA announces the prefix to its iBGP peers, that is, route reflectors RC and RD.
- Route reflectors RC and RD, announce the prefix to its iBGP client peers (RB), and non-client peers (RE and RF), as well as to each other. When RC received the announcement from RD, and vice versa, the announcement is ignored since both route reflectors belong to the same cluster.
- Router RB announces the prefix to another BGP router in a remote AS, through an eBGP session.
- Route reflectors RE and RF receive the announcement from non-client peers, so they will announce the prefix to its clients only, that is, routers RG and RH.
- Routers RG and RH will announce the prefix to eBGP peers in a remote AS.

2.2.2 BGP Confederations

When an iBGP core is being scaled, the core could be divided into smaller routing domains, that is, in different AS. But splitting the core and creating several new AS, would modify the AS path of the prefixes that originate or that go across the splitted core. To solve this issue, and make all this set of AS look like a single AS from outside the core, **BGP Confederations** are used.

How Confederations Work

While iBGP sessions within a sub-AS contained in a BGP Confederation work in the same way as the iBGP sessions in an AS outside of a confederation, the eBGP sessions are a bit more complex.

The **AS Confederation Sequence** and **AS Confederation Set** attributes are used to make this distinctions within a confederation.

- AS Confederation Sequence attribute contains all the AS within a BGP Confederation a particular route has passed through, in the **order** in which the AS were traversed.
- AS Confederation Set, which is a list of all the AS within a confederation a particular update has passed through, in no particular order.

Note that the AS Confederation Set is used in the same way as the AS sets in normal BGP, that is, this attribute is used with aggregation in order to reduce the size of the AS Path information

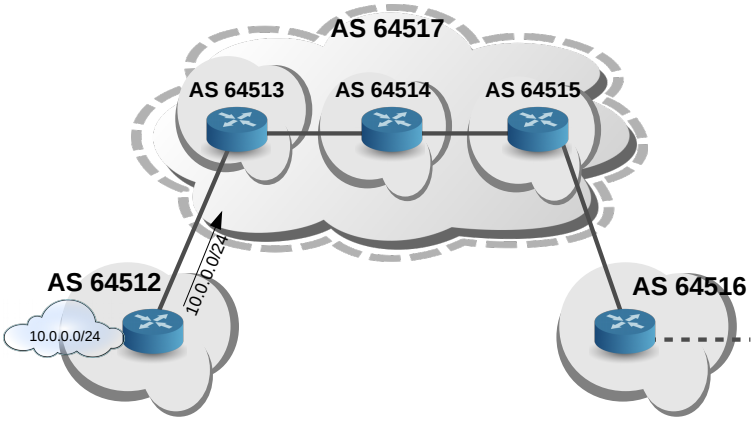


Figure 2.7: AS Confederation.

In Figure 2.7, route 10.0.0.0/24 is propagated through a confederation that contains AS 64513, 64514 and 64515. In this particular case, the AS path and the AS confederation set attributes of this prefix announcement will evolve as follows when advertised by the different routers:

1. AS 64512 router update:
    - AS Path: 64512
2. AS 64513 router update:
    - AS Path: 64512
    - AS Confederation Set: 64513
3. AS 64514 router update:
    - AS Path: 64512
    - AS Confederation Set: 64513, 64514
4. AS 64515 router update:
    - AS Path: 64512, 64517
    - AS Confederation Set: (**Attribute not announced in the update message**)
5. AS 64516 router update:
    - AS Path: 64512, 64517, 64516

The AS confederation formed by the three interior autonomous systems, is seen as the single AS 64517 for the following routers that receive the announcement of the route after going through the AS confederation

2.3 BGP Policies

In BGP routes' attributes can be modified depending on different parameters. Policies to modify the attributes' values, can be implemented in Quagga, by using route maps and prefix lists. Route maps can be used to filter routes by using prefix lists, Community Lists to apply common policies to groups of prefixes, AS Path Access Lists to filter prefixes based on its AS path, among others. Attribute modification in BGP peers is usually applied within route maps, that can be used to modify attributes depending on the neighbor from which the prefix was learned, or matching the prefix against a prefix list before applying a certain attribute modification.

2.3.1 Prefix Lists

Prefix Lists has been designed to filter routing information. When a prefix list is defined in Quagga, it follows the following syntax:

- # ip prefix-list *name* [permit | deny] *prefix*

The prefix list *name* permits or denies the announcement a certain prefix *prefix*.

Once a prefix list is defined, it can be applied to inbound and outbound prefixes from or to a certain neighbor in Quagga using the following syntax:

- # neighbor *peer* prefix-list *name* [in|out]

By using this syntax, a certain BGP router will apply the prefix list called *name* to its neighbor *peer* at the prefixes announced to the neighbor (out) or the prefixes announced by the neighbor (in).

An example of applying prefix lists to route filtering is the following one:

```
!
ip prefix-list list1 deny 203.0.113.0/24
ip prefix-list list1 permit 198.51.100.0/24
!

router bgp 64512

  neighbor 192.168.0.1 remote-as 64513

  neighbor 192.168.0.2 prefix-list list1 out

!
```

The configuration shown means the following behaviour:

Firstly, *list1* is defined.This prefix list will allow the prefix 198.51.100.0/24 and will deny 203.0.113.0/24. It has not been defined yet if the prefix list is applied to inbound or outbound traffic, and to which peer. These part of the configuration is done when the prefix list is set.

Once the prefix list has been defined, it is applied, in the case of BGP, to a certain neighbor, and specified if the prefix list is applied to the entering/exiting prefixes from/to a certain neighbor. In the example case, the prefix list *list1* will be applied to the prefixes announced to the neighbor 192.0.2.1. The behaviour will be, denying the announcement of all the prefixes within the 203.0.113.0/24 range, and permitting all the other prefix announcements within the 198.51.100.0/24 range.

2.3.2 BGP Route Maps

With route maps, route filter policies plus attribute manipulation can be applied together to routes received or sent to neighbors via update messages.

When a route map wants to be applied to a certain neighbor, the syntax is as follows:

- neighbor *peer* route-map *route-map-name* (in|out)

In the previous sentence, a route map called *route-map-name* will be applied to routes in the update messages announced or received by the neighbor *peer*, depending if it is set *out* or *in* value in the sentence, respectively.

For instance, the following route-map line would apply the route map called *testmap* to all the routes announced by the neighbor 10.0.0.1

```
router bgp 64512
 neighbor 10.0.0.1 remote-as 64513
 neighbor 10.0.0.1 route-map testmap in
!
.  .  .
```

Now it has been defined how to apply a route map to routes announced to/by a neighbor. The following lines explain how to define the route map in order to manipulate attributes and filter routes.

- route-map *route-map-name* (permit|deny) *order*

For each order, a set of set and match sentences can be set. If there is a mismatch, the next-order route map entry will be executed (note that the lower order is the first to execute). If the *n*th order route map entry matches, the route will be permitted or denied depending on what it is specified.

```
router bgp 64512
 neighbor 192.168.1.1 remote-as 64513
 neighbor 192.168.1.1 route-map TESTMAP in
!
ip prefix-list TESTLIST permit 10.0.0.0/8
!
route-map TESTMAP permit 10
 match ip address prefix-list TESTLIST
 set local-preference 150

route-map TESTMAP permit 20
 set ip next-hop 192.168.2.1
!
```

What the previous route map configuration does is the following:

1. Once a route update is received from neighbor 192.168.1.1, execute route map *TESTMAP*.
2. The first set of instructions are contained in the lowest order value for the route map called *TESTMAP*, that is, order 10. For instance, let's suppose that the network 10.0.1.0/24 is received in a BGP UPDATE message from neighbor 192.168.1.1. In this case, the match of 10.0.1.0/24 against the prefix list will be true, and the local preference will be set to 150 for this route. Then, no other route map set of instructions will be executed, since the order 10 set has matched.
3. In the case that, for instance, the prefix received does not match with the prefix list *TESTLIST*, the order 20 set will be executed, that is, the next hop value for the route will be set to 192.168.2.1.

Note that there are *match* and *set* commands. If there is no *match* command specified, further *set* commands are executed directly, since no *match* sentence is considered as a match.

Match Commands

The possible match commands that can be used within a route map, among others, are the following ones:

- Match the route against a previously-defined prefix list:

  match ip address prefix-list *prefix-list-name*

- Match the next hop value:

  match ip next-hop *ipv4_addr*

- Match a specified as-path:
  match aspath *as_path*
- Match a specified metric:
  match aspath *as_path*

Set Commands

The possible set commands that can be used within a route map, among others, are the following ones:

- Set a particular next hop value:
  set ip next-hop *ipv4_address*
- Set a local preference value:
  set local-preference *local_pref*
- Set the route's weight:
  set weight *weight*
- Set the MED value:
  set metric *metric*
- Set an AS path to prepend:
  set as-path prepend *as_path*

2.3.3 Policy Routing

Policies applied to BGP routing information, that is, to the prefixes and attributes contained within BGP updates, should not be confused with Policy Routing.

Policy Routing consists in forwarding packets, not only based on the destination address, but also checking different parameters of its headers. For instance, a packet could be send through different links depending on its source address, packet size or payload, among other parameters. There is a notable difference between applying BGP policies, and Policy Routing

Chapter 3

Practices

3.1 Introduction

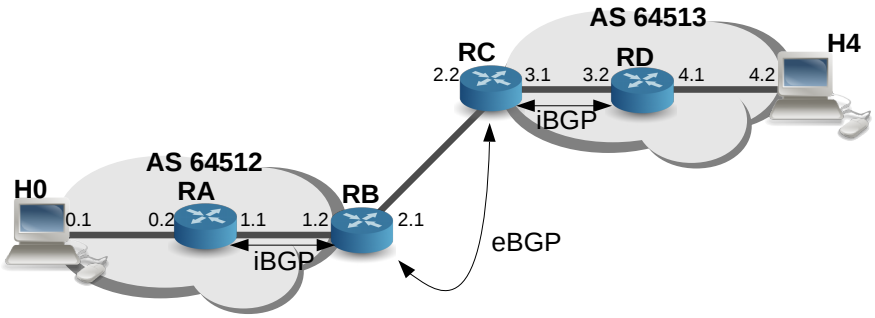


Figure 3.1: Basic scenario.

In the scenario in Figure 3.1, there are four networks: 10.0.0.0/24, 10.0.1.0/24, 10.0.2.0/24 and 10.0.3.0/24, also named as NetX, where X means 10.0.X.0/24.

Firstly, start the scenario and execute the initial configuration using the following commands:

```
# vnx -f first_bgp_scenario.xml -v --create
# vnx -f first_bgp_scenario.xml --execute initial
```

To configure a BGP peering session in a router, firstly you have to execute the *configure terminal* command in *Quagga*:

```
rx# configure terminal
```

Then, indicate that it is a BGP router, and the AS number where the router being configured belongs to, for instance, AS 12345:

```
rx(config)# router bgp 12345
```

To configure a peering session with a particular neighbor, for instance, with neighbor 1.2.3.4 that belongs to AS 45678, the following command has to be executed:

```
rx(config-router)# neighbor 1.2.3.4 remote-as 45678
```

1. Create an iBGP peering session among routers RA and RB. Use *Wireshark* to observe the negotiation process in Net1, and comment it.

2. What kind of information about network prefixes can you see in the UPDATE messages that the routers have exchanged, when the negotiation process has finished?

3. Establish the two remaining BGP sessions between routers RB and RC, and routers RC and RD.

4. At this point, all the necessary peering sessions should be working. You can check the neighbors in each BGP router by executing the following *Quagga* command:

```
rx# show bgp neighbors
```

5. Add a default route at each host in the scenario, and send a ping from host H0 to H4. Why it does not work?

   Check the BGP table, and the routing table in the scenario routers, using the following Quagga commands respectively:

```
rx# show ip bgp
rx# show ip route
```

   As you can see, the only networks in the routing table are the one where the router's interfaces belong to, that is, the directly connected networks. Distribute the appropriate networks through the BGP routers, from routers RA and RD.

6. Check the BGP table and the routing table in routers RA and RD. Why do you think the announced routes have not been installed in the routing tables? How could this problem be solved?

7. Add the routes to the next hop networks in routers RA and RD, and check the connectivity among the hosts (using the *route* command from *Linux*, not within *Quagga*).

3.2 Aggregation

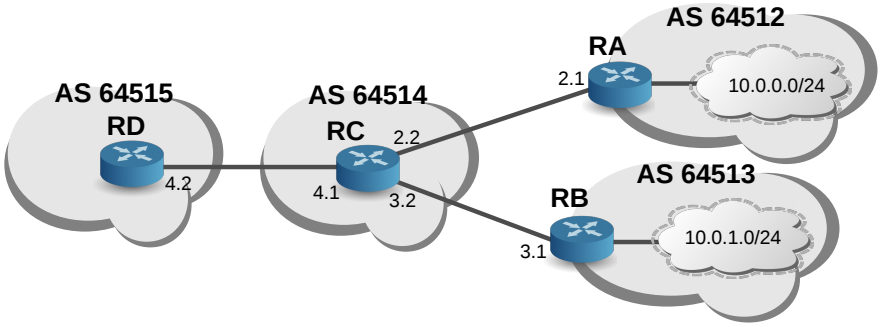


Figure 3.2: Aggregation practices scenario.

In the following scenario, different configurations for BGP aggregation are going to be on tested. Note that each network in the scenario NetX means 10.0.X.0/24. Firstly start the scenario in Figure 3.2 using the following command:

```
# vnx -f bgp_aggregation.xml -v --create
```

Then, execute the initial configuration to configure the BGP peering sessions among the routers:

```
# vnx -f bgp_aggregation.xml --execute initial
```

To aggregate addresses, the following *Quagga* command can be used, for instance:

```
rx(config-router)# aggregate-address 1.0.0.0/24
```

1. Announce networks 10.0.0.0/24 and 10.0.1.0/24 from routers RA and RB respectively.
2. Show the BGP table in router RD, to see if the previously announced routes appear.
3. Aggregate the routes in router RC, using the aggregate-address command, and explain what kind of information you can observe in the Net4 update message and the router RD BGP table.
4. Now, withdrawn the aggregate from router RC, and comment the update message that appears in Net4. Then announce the aggregate again, but adding the summary only option. Explain the update messages you observe, and the resulting BGP table in router RD. Which is the function of the summary-only option?
5. Withdrawn again the aggregate in router RC, and announce it again using the *as-set* option. Explain the messages observed on Net4, and what contains the AS path.
6. Now withdrawn the prefix 10.0.1.0/24 from router RB. Explain what the update messages in Net4 contain. Is there any AS set now in the update message/s? Why?
7. Withdrawn the prefix 10.0.0.0/24 from router RA. Comment the router RD BGP table.

3.3 Route Reflectors

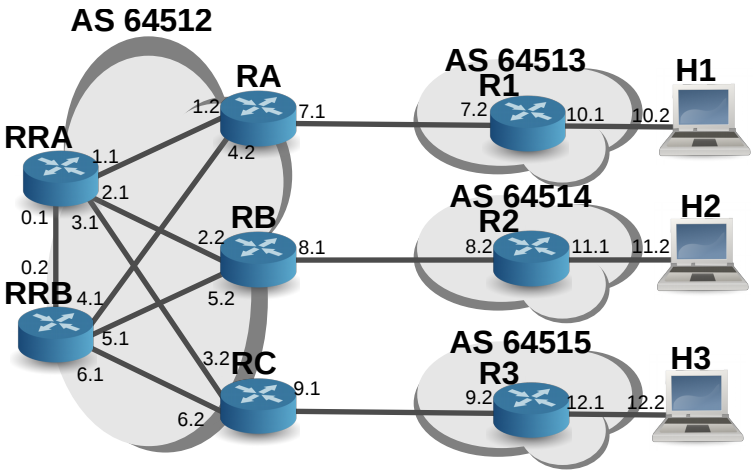


Figure 3.3: Route reflectors practices scenario.

In the scenario shown in Figure 3.3, connectivity among hosts H1, H2 and H3, want to be achieved. To do so, a proper configuration within all the routers in the scenario is required. The networks presented belong to the range 10.0.X.Y/24, where X is the scenario network name, that is, networks are called NetX.

1. Firstly, start the scenario using the following command:

```
# vnx -f bgp_route_reflectors.xml -v --create
```

Once it has started, execute the initial configuration, which initializes RIP protocol within the AS 64512, and established default routes for the hosts. Note that this will make the routers in AS 64512 know the inter-AS networks, that is, Net7, Net8, and Net9.

```
# vnx -f bgp_route_reflectors.xml --execute initial
```

2. Configure the necessary BGP sessions to achieve BGP connectivity among autonomous system 64512, and the other autonomous systems.

3. Configure router RRA as a route reflector, and make RA its route reflector client.

4. Configure two iBGP sessions: one among routers RRA and RB, and another one among routers RRA and RC.

5. Distribute networks Net10, Net11 and Net12 from routers R1, R2 and R3 respectively.

6. Try connectivity within hosts using the ping command. Are the results what you expected? Explain the behaviour observed.

7. Achieve connectivity among hosts H2 and H3, configuring only routers RB and RC. Explain how can you do it.

8. Is there any other way, by configuring router RRA only, and disabling the previous configurations in routers RB and RC, to achieve connectivity among hosts H2 and H3?

9. Make routers RA, RB and RC, route reflector clients of router RRA. Do the same for RRB, and also establish an iBGP session among routers RRA and RRB. The number of iBGP sessions within AS 64512 after the configuration should be 7. Note that the purpose of router RRB in this scenario is maintaining connectivity in the case one of the two route reflector fails, that is, it provides redundancy.

10. Observe the BGP tables of the edge routers (RA, RB and RC) in AS 64512 (show ip bgp). What can you observe about the routes learned through route reflectors? What problem may occur if more and more prefixes were advertised from outside into AS 64512?

11. Make a proper configuration to avoid route duplicity, by configuring the route reflectors properly.

3.4 Prefix Lists

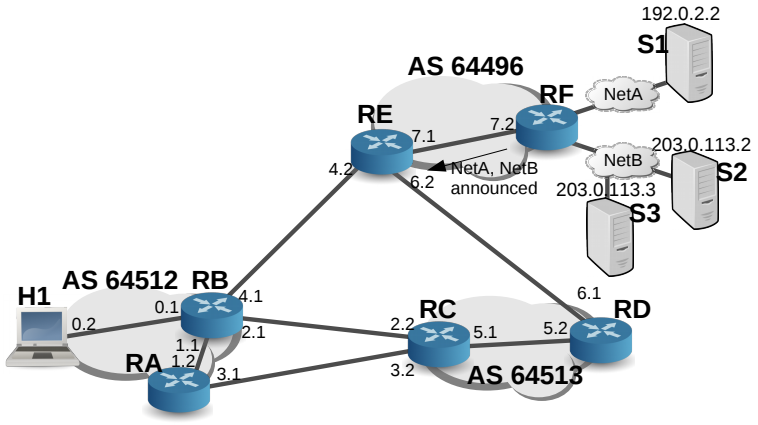


Figure 3.4: Prefix lists practices scenario.

In Figure 3.4, in networks 192.0.2.0/24 (NetA) and 203.0.113.0/24 (NetB) are placed servers S1, S2 and S3. The other networks presented belong to the range 10.0.X.Y/24.

Firstly, execute these two commands to start the scenario and to start the BGP peering sessions among the routers, respectively:

```
# vnx -f bgp_prefix_lists.xml -v --create
# vnx -f bgp_prefix_lists.xml --execute initial
```

1. Announce prefixes belonging to NetA and NetB from router RF(networks 192.0.2.0/24 and 203.0.113.0/24 respectively).

2. Do the same for the network where host H1 belongs to, announcing it from router RB.

3. Check connectivity between the host and the servers using the ping command. Comment which is the route the ping follows. Which do you think is the decisive criteria to route packets through that link?

4. Apply an appropriate prefix list within router RE so that ping request messages to 203.0.113.0/24 network now flow through 10.0.6.0/24 network. Note that even after the prefix-list is applied, the routes within AS 64512 routers will not vary until the prefix 203.0.113.0/24 is withdrawn and announced again.Then, withdrawn and announce again to see that the prefix-list takes effect and the prefix is not announced anymore through network 10.0.4.0/24

5. As you can see, the ping request/reply messages do not follow the same route. Add the appropriate prefix list in router RE so that ping request/reply messages flow through the same networks. Again, withdrawn and then announce network 10.0.0.0/24 from router RB to see if the prefix lists are properly applied.

3.5 Redistribution

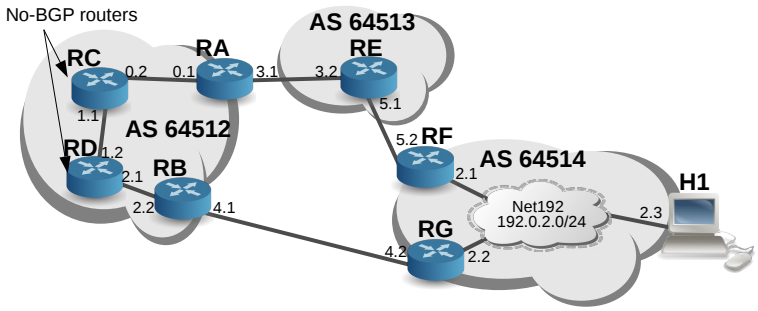


Figure 3.5: Redistribution scenario.

In the scenario shown in Figure 3.5, there is the RIP protocol running in all the AS 64512's routers. All the routers in the scenario except from routers RA and RB, run the BGP protocol.

Firstly, execute the following commands and to configure BGP and RIP protocol in the scenario's routers.

```
# vnx -f bgp_redistribution.xml -v --create
# vnx -f bgp_redistribution.xml --execute initial
```

1. Firstly, announce the prefix 192.0.2.0/24 from router RG.
2. Check the BGP tables in routers RA and RB. Why there are two entries in RA's table, and one in router RB's table?
3. Turn off interface *eth2* from router RB. Comment the process and BGP messages that appear at Net0, and the BGP tables in these two routers. Once you have finished this question, turn on eth2.
4. Redistribute the BGP network (or prefixes) into RIP protocol, in routers RA and RB. To do so, follow this sample commands:

   ```
   rx# configure terminal
   rx(config)# router rip
   rx(config-router)# redistribute bgp
   ```

   Tap Net0 using *Wireshark*, send a ping message from router RC to host H1 explain what happens, and how can it be solved. Checking the routing tables in routers RA and RC can help.
5. Apply the solution/s you have found to solve this routing loop.

3.6 Load Balancing and Attribute Manipulation

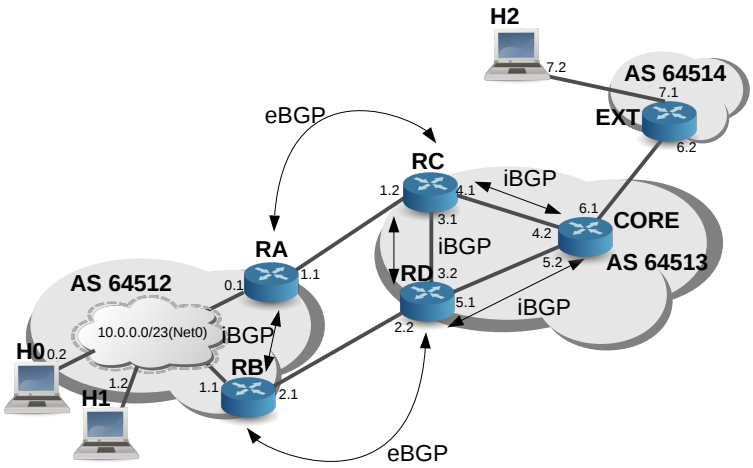


Figure 3.6: Load balancing scenario.

The scenario in Figure 3.6, contains three autonomous systems. Routers in AS 64513 run BGP and RIP protocols. Note that RIP protocol is used to reach the next hop ip addresses when a BGP network is announced through this AS. All networks NetX belong to the range 192.168.X.0/24, except for Net0, that is the 10.0.0.0/23 network.

Firstly, execute the following commands to create the scenario and to configure the BGP sessions and RIP protocol among all the routers in the scenario:

```
# vnx -f bgp_load_balancing.xml -v --create
# vnx -f bgp_load_balancing.xml --execute initial
# vnx -f bgp_load_balancing.xml --execute full-config
```

It is important to remember the BGP best path decision algorithm for this practices scenario.

1. Announce network 10.0.0.0/23 from routers RA and RB. Then, check the router CORE BGP table to see which of the two routes has been selected by the BGP best path decision algorithm. Capture the BGP update messages in networks 192.168.4.0/24 and 192.168.5.0/24 to see the attribute values.

2. Build a route map named *map1* in router RA to make the route through this router less preferable, by modifying the Multi Exit Discriminator (MED) value when announcing routes to router RC. By doing this, the intention is controlling the inbound traffic to AS 64512 so that it flows through Net1. Remember that the lower the MED value is, the most preferable a route is, and the default MED value is 0.

3. Apply the previous route map to the announcements made to neighbor 192.168.1.2, withdrawn the network from both routers RA and RB. Then, announce the network 10.0.0.0/23 from router RB, and then from router RA after 30 seconds. Capture the BGP update messages in Net3 and Net4. Is there any update message announced by router RA to their neighbors? Why? Send ping icmp packets from hosts H0 or H1 to host H2, in order to check that the route the packets follow is coherent. Note that Net7 has to be announced from router EXT before trying any ping ICMP message.

4. At this point, create a route map in router RB called *map2* to prepend autonomous systems to the AS path, so that it becomes longer than the AS path announced by router RA. Prepend, for instance, the AS 64512, so that it will appear two times in the AS path. This route map could be used, for instance, to try controlling the AS 64512 inbound traffic if the MED value is reseted in AS 64513 edge routers.

5. Apply route map *map2* to router RB eBGP peer, withdrawn routes from routers RA and RB, and announce these routes again. Check the BGP tables in AS 64513 routers, and comment the results. Is that what should be

expected? What is the relation with the BGP best path selection algorithm? Check the configuration by sending ping ICMP packets to host H2.

6. In the previous questions, different techniques to control inbound load balancing from AS 64512 perspective have been tested. Now, route maps are going to be applied in AS 64513 routers to control the outbound traffic flow of this AS.

   Create a route map *map3* in router RD to set a local preference higher than 100, for routes announced by router RB. Then apply this route map to its BGP peer router RB, withdrawn and announce route 10.0.0.0/23 again from routers RA and RB. What is the effect in AS 64513 routers?

7. At this point,the inbound traffic flow to AS 64512 will flow through Net2, since the local preference value has made route through router RB more preferable for AS 64513 routers. That means that, if AS 64513 is not under our control, the inbound traffic flow in AS 64512 cannot be controlled by manipulating attributes such as MED or AS path length, since the local preference value is preferable over the other attribute comparison. Since hosts H0 and H1 are distributed within 10.0.0.0/24 and 10.0.1.0/24 networks, how this fact could be used to balance the inbound traffic flow, if these two hosts have about the same volume of inbound traffic? Apply a solution in routers RA and RB. Check that ping replies flow through Net1 and Net2 to hosts H0 and H1 respectively.

Chapter 4

Results

In this chapter, the results of the practices' scenarios are included.

4.1 Aggregation

1. Announce networks 10.0.0.0/24 and 10.0.1.0/24 from routers RA and RB respectively.

```
ra# configure terminal
ra(config)# router bgp 64512
ra(config-router)# network 10.0.0.0/24

rb# configure terminal
rb(config)# router bgp 64513
rb(config-router)# network 10.0.1.0/24
```

2. Show the BGP table in router RD, to see if the previously announced routes appear.

```
rd# show ip bgp
```

   The routes appear, as expected

3. Aggregate the routes in router RC, using the aggregate-address command, and explain what kind of information you can observe in the Net4 update message and the router RD BGP table.

```
% rc# configure terminal
% rc(config)# router bgp 64514
% rc(config-router)# aggregate-address 10.0.0.0/23
```

   The update message contains AS 64512 and AS 64513 within the AS Set attribute. The AS Path attribute just contains AS 64514, as expected.

4. Now, withdrawn the aggregate from router RC, and comment the update message that appears in Net4. Then announce the aggregate again, but adding the summary only option. Explain the update messages you observe, and the resulting BGP table in router RD. Which is the function of the summary-only option?

```
rc(config-router)# no aggregate-address 10.0.0.0/23
```

   Update message withdrawing the aggregate, the two original prefixes are not withdrawn.

```
rc(config-router)# aggregate-address 10.0.0.0/23 summary-only
```

When the summary-only option in announced, the first update messages withdrawn the two /24 prefixes. Then 10.0.0.0/23 prefix is announced in the second update message. The function of the summary only option does not advertised the prefixes contained within the aggregate.

5. Withdrawn again the aggregate in router RC, and announce it again using the *as-set* option. Explain the messages observed on Net4, and what contains the AS path.

   When announcing the aggregate using the as-set option, the AS path is now AS 64514 plus an AS set that contains 64513 and 64512. The AS set contains a group of autonomous systems in a no particular order, and announcing the three prefixes outside the AS set would be incorrect since it would mean that reaching AS 64512 should be done through AS 64513 and vice versa.

6. Now withdrawn the prefix 10.0.1.0/24 from router RB. Explain what the update messages in Net4 contain. Is there any AS set now in the update message/s? Why?

   An update message containing the withdrawn of the prefix is sent from router RC to router RD, then, another update message containing the aggregate is sent, but in this case the AS path contains AS 64514 and AS 64512, with no AS set. In this case AS set is not necessary, since the AS path of the aggregate is strictly correct after the prefix 10.0.1.0/24 withdrawal.

7. Withdrawn the prefix 10.0.0.0/24 from router RA. Comment the router RD BGP table.

   When both prefixes from routers RA and RB are withdrawn, the aggregate is also withdrawn from router RC since there are no other prefixes contained within the aggregate being advertised by other routers.

4.2 Route Reflectors

1. Firstly, start the scenario using the following command:

```
# vnx -f bgp_route_reflectors.xml -v --create
```

   Once it has started, execute the initial configuration, which initializes RIP protocol within the AS 64512, and established default routes for the hosts. Note that this will make the routers in AS 64512 know the inter-AS networks, that is, Net7, Net8, and Net9.

```
# vnx -f bgp_route_reflectors.xml --execute initial
```

2. Configure the necessary BGP sessions to achieve BGP connectivity among autonomous system 64512, and the other autonomous systems.

```
Establishing eBGP sessions:

r1# configure terminal
r1(config)# router bgp 64513
r1(config-router)# neighbor 10.0.7.1 remote-as 64512
---
ra# configure terminal
ra(config)# router bgp 64512
ra(config-router)# neighbor 10.0.7.2 remote-as 64513
---
r2# configure terminal
r2(config)# router bgp 64514
r2(config-router)# neighbor 10.0.8.1 remote-as 64512
---
rb# configure terminal
rb(config)# router bgp 64512
rb(config-router)# neighbor 10.0.8.2 remote-as 64514
---
r3# configure terminal
```

```
r3(config)# router bgp 64515
r3(config-router)# neighbor 10.0.9.1 remote-as 64512
---
rc# configure terminal
rc(config)# router bgp 64512
rc(config-router)# neighbor 10.0.9.2 remote-as 64515
```

3. Configure router RRA as a route reflector, and make RA its route reflector client.

   First, starting an iBGP session:

```
ra(config-router)# neighbor 10.0.1.1 remote-as 64512
```

   Configuring the route reflector:

```
rra(config-router)# neighbor 10.0.1.2 route-reflector-client
```

4. Configure two iBGP sessions: one among routers RRA and RB, and another one among routers RRA and RC.

```
rra(config-router)# neighbor 10.0.2.2 remote-as 64512
rra(config-router)# neighbor 10.0.3.2 remote-as 64512
---
rb(config-router)# neighbor 10.0.2.1 remote-as 64512
---
rc(config-router)# neighbor 10.0.3.1 remote-as 64512
```

5. Distribute networks Net10, Net11 and Net12 from routers R1, R2 and R3 respectively.

```
r1(config-router)# network 10.0.10.0/24
---
r2(config-router)# network 10.0.11.0/24
---
r3(config-router)# network 10.0.12.0/24
```

6. Try connectivity within hosts using the ping command. Are the results what you expected? Explain the behaviour observed.

   Connectivity among routers 2 and 3 since these two routers do not know the nets 12 and 11 respectively. When these two nets are announced to the route reflector RRA, the route reflector proceeds in the following way:

   Both networks 11 and 12 are learned through an iBGP non-client router, that is, routers RB and RC respectively. Since these prefixes are received from a non-client peer, the route reflector announces the prefix to its clients only, that is, to RA. On the other hand connectivity among host H1 and the other hosts is possible since the route reflector learns the prefix from a client. When a prefix is learned from a client, it is advertised to both client and non-client peers, so network 10.0.10.0/24 can be learn by both routers RB and RC, and then announced through its respective iBGP sessions to routers R2 and R3.

7. Achieve connectivity among hosts H2 and H3, configuring only routers RB and RC. Explain how can you do it.

   Establishing an iBGP session among routers rb and rc will make it work. Since the route reflector does not reflect a prefix learn from a non-client peer to other non-client peers, a way to achieve connectivity among networks 11 and 12 is establishing an iBGP peering session among routers RB and RC.

```
rb# configure terminal
rb(config)# router bgp 64512
rb(config-router)# neighbor 10.0.3.2 remote-as 64512
---
rc# configure terminal
rc(config)# router bgp 64512
rc(config-router)# neighbor 10.0.2.2 remote-as 64512
```

8. Is there any other way, by configuring router RRA only, and disabling the previous configurations in routers RB and RC, to achieve connectivity among hosts H2 and H3?

```
Making RB and RC route reflector clients of RRA.
---
rra(config-router)# neighbor 10.0.2.2 route-reflector-client
rra(config-router)# neighbor 10.0.3.2 route-reflector-client
```

9. Make routers RA, RB and RC, route reflector clients of router RRA. Do the same for RRB, and also establish an iBGP session among routers RRA and RRB. The number of iBGP sessions within AS 64512 after the configuration should be 7. Note that the purpose of router RRB in this scenario is maintaining connectivity in the case one of the two route reflector fails, that is, it provides redundancy.

```
rra# configure terminal
rra(config)# router bgp 64512
rra(config-router)# neighbor 10.0.0.2 remote-as 64512
---
rrb(config)# router bgp 64512
rrb(config-router)# neighbor 10.0.0.1 remote-as 64512
rrb(config-router)# neighbor 10.0.4.2 remote-as 64512
rrb(config-router)# neighbor 10.0.5.2 remote-as 64512
rrb(config-router)# neighbor 10.0.6.2 remote-as 64512
rrb(config-router)# neighbor 10.0.4.2 route-reflector-client
rrb(config-router)# neighbor 10.0.5.2 route-reflector-client
rrb(config-router)# neighbor 10.0.6.2 route-reflector-client
---
ra(config)# router bgp 64512
ra(config-router)# neighbor 10.0.4.1 remote-as 64512
ra(config-router)#
---
rb# configure terminal
rb(config)# router bgp 64512
rb(config-router)# neighbor 10.0.5.1 remote-as 64512
---
rc# configure terminal
rc(config)# router bgp 64512
rc(config-router)# neighbor 10.0.6.1 remote-as 64512
```

10. Observe the BGP tables of the edge routers (RA, RB and RC) in AS 64512 (show ip bgp). What can you observe about the routes learned through route reflectors? What problem may occur if more and more prefixes were advertised from outside into AS 64512?

    The prefixes learned through route reflectors appear twice, one learned from router RRA and the other learned from RRB. Route duplicity might cause a resource exhaustion problem in edge routers, so route reflector attributes could be used to solve this issue.

11. Make a proper configuration to avoid route duplicity, by configuring the route reflectors properly.

    By setting both route reflectors in the same reflection cluster, when clients receive the route duplicated, they just ignore it since they can see that it belongs to the same reflection cluster that the one they have. This configuration might be even more important if there are several redundant route reflectors and a great amount of announced prefixes.

    For instance, it is set 10.0.0.2 as the cluster ID.

```
rra# configure terminal
rra(config)# router bgp 64512
rra(config-router)# bgp cluster-id 10.0.0.2
---
rrb# configure terminal
rrb(config)# router bgp 64512
rrb(config-router)# bgp cluster-id 10.0.0.2
```

4.3 Prefix Lists

1. Announce prefixes belonging to NetA and NetB from router RF(networks 192.0.2.0/24 and 203.0.113.0/24 respectively).

```
rf# configure terminal
rf(config)# router bgp 64496
rf(config-router)# network
rf(config-router)# network 203.0.113.0/24
```

2. Do the same for the network where host H1 belongs to, announcing it from router RB.

```
rb# configure terminal
rb(config)# router bgp 64512
rb(config-router)# network 10.0.0.0/24
```

3. Check connectivity between the host and the servers using the ping command. Comment which is the route the ping follows. Which do you think is the decisive criteria to route packets through that link?

   The ping messages flow through network 10.0.4.0/24 since the alternative routes go through AS 64513, so that the AS path is one AS longer. When the BGP best path decision algorithm is applied, since the higher preference attributes are equal, the AS path length is the decisive factor.

4. Apply an appropriate prefix list within router RE so that ping request messages to 203.0.113.0/24 network now flow through 10.0.6.0/24 network. Note that even after the prefix-list is applied, the routes within AS 64512 routers will not vary until the prefix 203.0.113.0/24 is withdrawn and announced again.Then, withdrawn and announce again to see that the prefix-list takes effect and the prefix is not announced anymore through network 10.0.4.0/24

```
re# configure terminal
re(config)# ip prefix-list list1 deny 203.0.113.0/24
re(config)# router bgp 64496
re(config-router)# neighbor 10.0.4.1 prefix-list list1 out
```

5. As you can see, the ping request/reply messages do not follow the same route. Add the appropriate prefix list in router RE so that ping request/reply messages flow through the same networks. Again, withdrawn and then announce network 10.0.0.0/24 from router RB to see if the prefix lists are properly applied.

```
re# configure terminal
re(config)# ip prefix-list list2 deny 10.0.0.0/24
re(config)# router bgp 64496
re(config-router)# neighbor 10.0.4.1 prefix-list list2 in
---
rb(config-router)# no network 10.0.0.0/24
rb(config-router)# network 10.0.0.0/24
```

4.4 Redistribution

1. Firstly, announce the prefix 192.0.2.0/24 from router RG.

```
rg# configure terminal
rg(config)# router bgp 64514
rg(config-router)# network 192.0.2.0/24
```

2. Check the BGP tables in routers RA and RB. Why there are two entries in RA's table, and one in router RB's table?

   Router RA receives the prefix announcement from router re at all times, as well as the announcement from its iBGP peer router RB. From these two prefixes, the selected one by the BGP best path decision algorithm is through router RB. If, for any reason, router RA had announced the prefix received from router RE (long AS path) to its iBGP peer, router RA would withdrawn the long AS path route sending an UPDATE message to router RB, as soon as router RA receives an UPDATE message containing the same prefix but with a short AS path from router RB. That is, after running the BGP best path algorithm in router RA, the best path is the one announced by its iBGP peer. To sum up, if the prefix from router RB is received before the prefix announced by router RE, there is no need to withdrawn the long AS path prefix by router RA from router RB. Otherwise, the prefix must be withdrawn by router RA since the best route is through router RB.

3. Turn off interface *eth2* from router RB. Comment the process and BGP messages that appear at Net0, and the BGP tables in these two routers. Once you have finished this question, turn on *eth2*.

   When *eth2* is turned off, router RB automatically withdrawns the prefix received by router RG from BGP. When this withdrawal reaches router RA, the best (and only) path to the prefix in router RA is the one received from router RE. So router RA announces the long AS path prefix to router RB.

4. Redistribute the BGP network (or prefixes) into RIP protocol, in routers RA and RB. To do so, follow this sample commands:

```
rx# configure terminal
rx(config)# router rip
rx(config-router)# redistribute bgp
```

   Tap Net0 using *Wireshark*, send a ping message from router RC to host H1 explain what happens, and how can it be solved. Checking the routing tables in routers RA and RC can help.

   A loop is created, since the routes have been redistributed by the two edge routers in AS 64512. When router RC sends the ping message, the best path to Net192 is through router RA. On the other hand, router RA considers that the best route to Net192 is through router RB, so it forwards it back through the same interface where it received the ping message, to router RC.

   It could be solved by setting a physical link among routers RA and RB, or by disabling the iBGP peering session among these two routers. A third option could be setting in router RA, a higher weight value for its neighbor router RE. This would make the BGP best path decision algorithm choose the route through router RE.

5. Apply the solution/s you have found to solve this routing loop.

   Apply the previous question's solutions.

4.5 Load Balancing and Attribute Manipulation

1. Announce network 10.0.0.0/23 from routers RA and RB. Then, check the router CORE BGP table to see which of the two routes has been selected by the BGP best path decision algorithm. Capture the BGP update messages in networks 192.168.4.0/24 and 192.168.5.0/24 to see the attribute values.

   Since the attributes used in the BGP best path decision algorithm are equal (MED, AS PATH, Local Preference), the selected route through router RA is selected based on the lowest router ID value of the routers that announced the network.

```
ra# configure terminal
ra(config)# router bgp 64512
ra(config-router)# network 10.0.0.0/23
---
rb# configure terminal
rb(config)# router bgp 64512
rb(config-router)# network 10.0.0.0/23
```

2. Build a route map named *map1* in router RA to make the route through this router less preferable, by modifying the Multi Exit Discriminator (MED) value when announcing routes to router RC. By doing this, the intention is controlling the inbound traffic to AS 64512 so that it flows through Net1. Remember that the lower the MED value is, the most preferable a route is, and the default MED value is 0.

```
ra# configure terminal
ra(config)# route-map map1 permit 10
ra(config-route-map)# set metric 10
```

3. Apply the previous route map to the announcements made to neighbor 192.168.1.2, withdrawn the network from both routers RA and RB. Then, announce the network 10.0.0.0/23 from router RB, and then from router RA after 30 seconds. Capture the BGP update messages in Net3 and Net4. Is there any update message announced by router RA to their neighbors? Why? Send ping icmp packets from hosts H0 or H1 to host H2, in order to check that the route the packets follow is coherent. Note that Net7 has to be announced from router EXT before trying any ping ICMP message.

```
ra(config-router)# neighbor 192.168.1.2 route-map map1 out
```

   When the network is announced from router RA, the announcement reaches router RD with MED value 10. At this point, router RC has two routes to 10.0.0.0/23 network, and applies the BGP best path decision algorithm and selects the route from router RB, due to the MED value.

   When compares the MED values of the two possible routes, the lowest value (0) is the one through router RB, so the route announced by router RA, is not announced by router RC to its neighbor. If the routes were announced in the opposite order, that is, firstly from router RA, once it received the update message from the route originated by router RB, router RC would send update messages withdrawing the route originated by router RA.

4. At this point, create a route map in router RB called *map2* to prepend autonomous systems to the AS path, so that it becomes longer than the AS path announced by router RA. Prepend, for instance, the AS 64512, so that it will appear two times in the AS path. This route map could be used, for instance, to try controlling the AS 64512 inbound traffic if the MED value is reseted in AS 64513 edge routers.

```
rb# configure terminal
rb(config)# route-map map2 permit 10
rb(config-route-map)# set as-path prepend 64512
```

5. Apply route map *map2* to router RB eBGP peer, withdrawn routes from routers RA and RB, and announce these routes again. Check the BGP tables in AS 64513 routers, and comment the results. Is that what should be expected? What is the relation with the BGP best path selection algorithm? Check the configuration by sending ping ICMP packets to host H2.

```
rb(config-router)# neighbor 192.168.2.2 route-map map2 out
```

   When routes are re-advertised from routers RA and RB, the selected route by AS 64513 routers is the one through router RA, since the route through router RB has a longer AS path. Since the AS path length comparison has more preference than the MED comparison, in this case the selected route is the one that was previously withdrawn.

6. In the previous questions, different techniques to control inbound load balancing from AS 64512 perspective have been tested. Now, route maps are going to be applied in AS 64513 routers to control the outbound traffic flow of this AS.

   Create a route map *map3* in router RD to set a local preference higher than 100, for routes announced by router RB. Then apply this route map to its BGP peer router RB, withdrawn and announce route 10.0.0.0/23 again from routers RA and RB. What is the effect in AS 64513 routers?

```
rd(config)# route-map map3 permit 10
rd(config-route-map)# set local-preference 150
 ---
rd(config-router)# neighbor 192.168.2.1 route-map map3 in
```

Now, the preferred route to forward traffic to network 10.0.0.0/23 is through router RB since the route has a local preference value higher than the value through router RA. At this point, the two routes are only present in router RC's BGP table, since router RC withdrawns the route through router RA from its AS 64513 iBGP peers. in the other routers in AS 64513, the only route to 10.0.0.0/23 is the one through router RB, in a similar way as happened in the previous questions.

7. At this point,the inbound traffic flow to AS 64512 will flow through Net2, since the local preference value has made route through router RB more preferable for AS 64513 routers. That means that, if AS 64513 is not under our control, the inbound traffic flow in AS 64512 cannot be controlled by manipulating attributes such as MED or AS path length, since the local preference value is preferable over the other attribute comparison. Since hosts H0 and H1 are distributed within 10.0.0.0/24 and 10.0.1.0/24 networks, how this fact could be used to balance the inbound traffic flow, if these two hosts have about the same volume of inbound traffic? Apply a solution in routers RA and RB. Check that ping replies flow through Net1 and Net2 to hosts H0 and H1 respectively.

```
ra(config-router)# network 10.0.0.0/24
rb(config-router)# network 10.0.1.0/24
```

Announcing more specific routes means that there is no need to run the BGP best path decision algorithm since each route is more specific than the original /23 prefix, and each one is received by a single edge router in AS 64513.

Bibliography

[1] Russ White, Danny McPherson, and Srihari Sangli. *Practical BGP*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.

[2] Bassam Halabi. *Internet Routing Architectures*. Cisco Press, 1997.

[3] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard), January 2006. Updated by RFCs 6286, 6608, 6793, 7606, 7607, 7705.

[4] Quagga routing software suite. http://www.nongnu.org/quagga/. Accessed: 2017-01-10.

Yeray Sainz Lorenzo

Appendix B - Intermediate System to Intermediate System (IS-IS)

Contents

Chapter 1

Introduction

IS-IS (Intermediate System to Intermediate System) is a link-state routing protocol is an Interior Gateway Protocol (IGP), that is, thought to be run within an administrative domain. The IS-IS protocol is one of the most used protocols within the network backbones of large service providers. A particularity IS-IS has, compared to other IGPs, such as OSPF, is that it uses the OSI protocol stack, instead of the commonly used TCP/IP model. Using the OSI model means having a different jargon than the used in the TCP/IP model.

1.1 Main Features

IS-IS runs over the Layer 2 of the OSI model, which enables it to support other routing protocols over it. Unlike other IGPs such as OSPF, that is a Layer 3 protocol, it does not use IP to carry routing information. This makes IP addressing at interfaces not necessary.

As well as OSPF, it uses the Dijkstra algorithm to compute the best path to a certain destination, and the network topology is made out of areas. However, there are notable differences. While in OSPF the elements that separate areas are routers (known as Area Border Routers), in IS-IS, areas are separated by links, as shown in Figure 1.1. The links can belong to two different levels: Level 1, Level 2, or both.

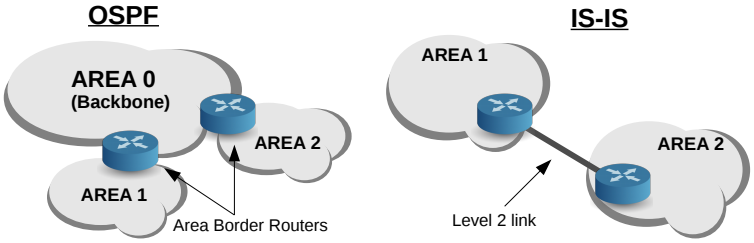


Figure 1.1: OSPF vs IS-IS comparison

Level 1 links are used for intra-area adjacencies (links to connect two routers belonging to the same area), while Level 2 links are used for inter-area adjacencies (links to connect two routers belonging to different areas). As well as that, there is not an specific backbone area, as happens with the OSPF "Area 0".

Regarding the route calculation, routers store the link-state information in databases known as Link State Database (LSDB). From the information stored is then applied the SPF algorithm to find the optimal path.

Another important feature compared to OSPF protocol is that IS-IS generates less link-state information packets per router. That is, all the information of a router's links is contained in a single link-state packet (called Link-State PDU in IS-IS)

1.2 IS-IS Addressing

OSI addresses are used in IS-IS for identifying nodes. Even though this addressing model is quite different regarding the semantics and the addressing paradigm, it is more simple than the IP addressing model.

1.2.1 The OSI Addressing Model

In the OSI addressing model, each OSI router, that is, a router running the IS-IS protocol, just needs an address configured. Note that commonly, in the IP addressing model, each interface in the router has an IP address assigned. By contrast, in the OSI addressing model, a single OSI address is configured for each router. This address can be set to the routing process or to the loopback address. The OSI address is also known as the Network Entity Title (NET), and the physical interfaces interfaces in a router can be left as unnumbered (not having an IP address configured), since it is a protocol that runs in the level 2 of the OSI protocol layer.

Network Entity Title (NET)

The Network Entity Title is composed of the following parts:

- **Area-ID**: The Area-ID field ranges from 1 to 13 bytes of length. Its common length is 1, 3 or 5 bytes. The first byte is called the Address Family Identifier (AFI), which indicates what type of address is being defined. For instance, when private NETs are defined, the AFI used for private addressing is 49. This is equivalent to IPv4 private networks (10.0.0.0/8, 172.16.0.0/12 and 192.168.0.0/16). The other bytes in this field (up to 12 bytes more), are used to identify the area the router belongs to.
- **System-ID**: The System-ID is a fixed 6-byte field, that must be unique for each router belonging to the network where IS-IS is running. The way system-IDs are assigned can be very different depending on the network administrators, that is, the allocation scheme is not common for all the networks where IS-IS is deployed. However, an easy way to obtain and assure system-IDs uniqueness, is from a loopback IP address translation. For instance, if the 192.168.1.2 IP address wants to be translated into a 6-byte system-ID, the result would be the following one: 1921.6800.1002

  The previous example has been made using the Binary-coded decimal (BCD) encoding. Each of the three numbers (192, 168, 1, and 2) has to become a three-digit number, if not, the first digits must be filled with zeros. Since 192 and 168 are already a three-digit numbers, are not modified. Then, 1 becomes 001, and 2 becomes 002. Finally the string of numbers 192168001001 is separated with dots every four digits (or two bytes), and the result is 1921.6800.1002
- **NSEL**: The Network Selector is the last byte of the NET. This byte must be always zero for IS-IS. If it is not set to zero (00), adjacencies will not form.

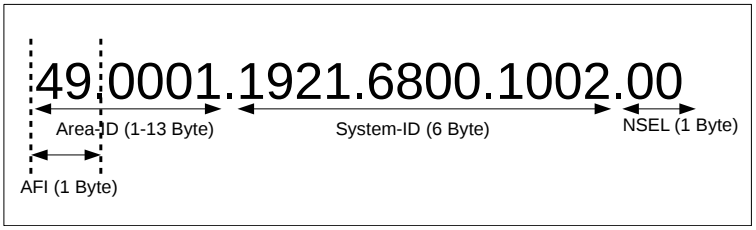


Figure 1.2: NET example

Now that all the fields of an OSI address or NET has been explained, a complete NET can be built. For instance, if a router with its loopback address being 192.168.2.1, that is within an IS-IS area 1, its NET could be the following one: **49.0001.1921.6800.1002.00**, as shown in Figure 1.2.

NETs are easier to understand when read from right to left, since the bytes on the left (in this case, the first three bytes 49.0001, that is, the Area-ID field) have a variable length (from 1 to 13 bytes).

Reading from the bytes on the right, it easy to identify the NSEL byte (00), and the 6-byte System-ID field as well (1921.6800.1002). Once these fixed-length fields are identified, the remaining bytes are part of the Area-ID field.

Focusing on the Area-ID, the first byte on the left is the AFI, which is 49 for private addressing, and the remaining bytes the area number the router belongs to.

1.3 IS-IS Areas and Levels

IS-IS network topology is made out of areas, and links that belong to a certain level (1, 2, or both), as previously introduced.

1.3.1 Areas

In IS-IS, areas are used to split the network so that the topological horizon of the routers does not grow up to a point that consumes a high amount of resources. Another capability of the areas, is the possibility to make network migrations smoothly without setting an specific maintenance gap time. As previously introduced, the border between two IS-IS areas is (at least) a Level-2 link between routers. Note that in OSPF, adjacencies among two routers cannot be formed if the interfaces in each side of the link do not belong to the same area. In IS-IS, there is no need to match the Area-ID in each side of the link to form an adjacency between two routers. However, as will be explained in the following subsection, links and routers belong to different levels to maintain different hierarchies and to separate areas.

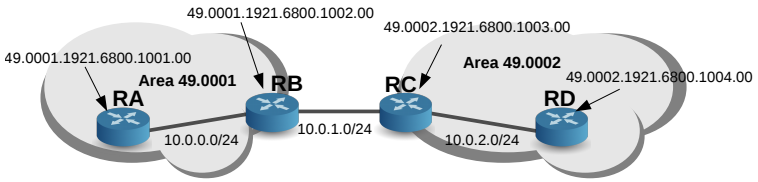


Figure 1.3: IS-IS area concept

In Figure 1.3, the scenario shown contains routers RA, RB, RC and RD. Before setting IS-IS in this network, and splitting the shown routers into two different areas, all the routers know the routes to 10.0.0.0/24, 10.0.1.0/24 and 10.0.2.0/24. When IS-IS is set in the scenario, routers RA and RD will contain the following routes in their routing tables: The directly connected networks (10.0.0.0/24 and 10.0.2.0/24 respectively), and a default route. The default route will have as gateways, routers RB and RC respectively. By doing two separate areas, routers RA and RD would just need a route to RB and RC respectively, to reach routers in the other area. On the other hand, routers RB and RC will contain the same routes in their routing tables as before.

1.3.2 Levels

While in OSPF the difference between areas is made by a "special" router (ABR), in IS-IS this distinction does not exist. The reason is that while the ABRs in OSPF belong to more than a single area at a time, while in IS-IS routers just belong to a single area. In OSPF, a pair of interfaces from different routers must belong to the same area to establish an adjacency, while in IS-IS it is not a necessary condition. The routing hierarchies are set from the topology levels. Each link can belong to Level 1, Level 2, or both levels at the same time.

- **Level 1**: Links belonging to Level 1, are the ones between two routers within the same area. For instance, in the previous Figure 1.3, the link between routers RA and RB, and the link between routers RC and RD, will belong to level 1, since both links connect routers of the same area.
- **Level 2**: Links belonging to Level 2 don't necessarily need to belong to the same area to establish adjacency, that is, it does not matter if the area ID's at both ends do not match. This would be the case of the link in Figure 1.3, between RB and RC.

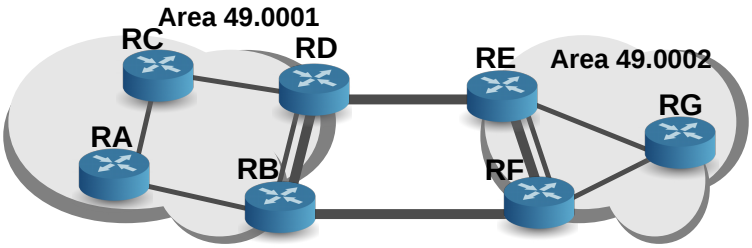


Figure 1.4: IS-IS level concept

The scenario shown in Figure 1.4, shows the different topology levels that exist between pairs of routers. On one hand, the thin links represent Level 1 topology links, and exist between routers belonging to the same area. On the other hand, the thick links represent the Level 2 links, that are set for the intra-area adjacencies. Note that there are some links that belong at the same time to Level 1 and Level 2 topologies at the same time.

1.3.3 Routing Hierarchy and Route Leaking

As seen in the previous scenario in Figure 1.4, the link between RB and RD, as well as the link between RE and RF belong simultaneously to levels 1 and 2. The reason of this, is the following rule: Routers sharing the same Area-ID, determine Level 1 topology, and circuits that share a contiguous set of Level 2 circuits determine the Level 2 topology.

Following this rule, that says that the Level 2 topology is contiguous, makes the existence of the links that belong to both topologies: the contiguous Level 2 topology, plus the Level 1 topology from their respective areas. The Level 1 topology is also present since both pairs of routers {RD,RB} and {RE,RF} belong to the same area.

If the Level-1 Level-2 links were Level-1 only, that is, the Level-2 contiguity rule would not be accomplished, the Level-1 Level-2 routers {RB,RD,RE,RF} would not have a complete link-state information about the whole topology. That is, routers RB and RF, would not know about the existence of the link between routers RD and RE. In the same way, routers RD and RE would not know about the existence of the link between routers RB and RF. Having just one Level-1 Level-2 link would be enough, since the contiguity of the Level-2 topology would be accomplished. However, by setting two Level-1 Level-2 links, in the case one of the two links stopped working, the contiguity rule would still be accomplished and Level-1 Level-2 routers would still have a complete link-state information. Note that the topology level a link belongs can be configured manually.

Regarding route leaking, IS-IS leaks routing information from the Level 1 topology to the Level 2 topology, but not in the opposite way. This behaviour makes all routers that just have Level 1 adjacencies not being flooded by routing information from other areas. The procedure for these routers, to route traffic to other areas, is by finding the shortest path to a router that has a Level 2 adjacency with another area.

Routers that have at least one Level 2 adjacency, must set a field called Attach bit (ATT) in the routing messages they send to the neighbour routers. When the routers belonging to the Level 1 topology only, receive a message from a router with the ATT set, those routers just calculate the shortest route to one of these routers, and a default route is set pointing to one of these Level 1 - Level 2 routers. This behaviour is what the Totally-Stubby-Areas do in the OSPF protocol.

1.4 SPF Computation

The idea under IS-IS route calculation, is distributing information of the network topology within the hierarchy levels separately. The information is contained in the Link State Packets (LSPs), and is stored in Link State Databases (LSDBs). There is a separate LSDB for Level 1 and for Level 2. Once the databases have received all the topology information, each router applies the SPF algorithm locally, to find out the shortest path to each destination in a finite number of steps.

Chapter 2

Fundamentals

2.1 Advertising Information Throughout the Network

In link-state protocols such as IS-IS or OSPF, information about network topology and IP reachability is distributed beyond the near neighbours. In IS-IS, this kind of information is sent within a Link State Protocol Data Unit (LSP). LSPs are generated and flooded in the network's routers, following a certain behaviour. IS-IS routers follow the distributed databases and local computation paradigm. That is, when a Link State Database (LSDB) is populated with topology information received from LSPs, the SPF algorithm is applied to find out which is the best path to destinations. There are two different databases, one for each Level 1 area, and another one for the Level 2.

2.1.1 Link State Protocol Data Unit

In order to populate the databases (LSDB) of the IS-IS routers in the network, the topology information is carried within a Link State Protocol Data Unit (LSP). LSPs are used to announce information such as the link states of the router, IP routes, checksum values among other parameters.

LSP Sequence Number

IS-IS routers can determine if information contained in an LSP should be updated in its LSDB from the sequence number. When a router receives an LSP from another router, the receiver will check in its database if the sequence number is greater than the last LSP received from the same router. If the sequence number is greater, the receiver will update the LSP entry in the database with the new information from the LSP. Otherwise, it will not update the information and the LSP will be discarded.

LSP Lifetime and Periodic Refresh

The validity of an LSP cannot last forever in the network router's databases. For instance, if a router is suddenly powered down, and does not send LSPs to its neighbours withdrawing the up links, the other routers in the network will not notice that links from/to the turned-off router are down, and will consider the links through this router valid for the SPF calculation. By setting a maximum validity time of LSPs (Lifetime), problems as the previously explained, can be avoided. If a router suddenly stops working, and does not advertise down links with an LSP, the LSPs installed in other routers' databases will be removed once the lifetime expires.

To avoid LSPs being removed from other routers' databases, LSPs must be originated periodically in a time minor than the LSP lifetime. The recommended refresh time, also known as origination interval, is the Lifetime minus 300 seconds.

Link State PDU Format

The LSP header is commonly fixed to 27 bytes, excluding the Layer 2 information such as MAC addresses. The most important elements in the LSP are the following ones:

1. **Lifetime**
2. **LSP-ID**
3. **Sequence Number**
4. **Checksum**

Lifetime and Sequence Number fields, have been previously explained. The Checksum value is used to make transmission errors recognizable, by checking that there is no imbalance between zeros and ones.

The LSP-ID is a fixed 8 byte field, that determines the LSP type(xxxx.xxxx.xxxx.yy-zz). The first 6 bytes are the System-ID value of the router that generated the LSP (xxxx.xxxx.xxxx). The following byte, represents the Pseudonode-ID (yy). What a pseudonode is, will be introduced in the following section. For now, just consider that if the Pseudonode-ID is 0, the LSP represents a real router. Otherwise, it represents a pseudonode. For now, just assume that in a LAN, there are real nodes (routers), and a pseudonode that represents the whole LAN. The last byte is the Fragment ID, that is used for fragmentation purposes (zz). Note that while in OSPF the fragmentation relies on IP, in IS-IS relies on itself since it is not IP-based.

2.1.2 TLVs

To encode information, such as IP reachability, into LSP or Hello messages, IS-IS uses a format called Type Length Value (TLV). The data encoded into each TLV, travels across the network to other IS-IS routers.

The structure of a TLV is composed of the following fields:

- **Type**: This is a 1-byte field used to determine the information type contained in the TLV. Each value of the type field, corresponds to a specific type of information. Type examples are:
  - #1 TLV: Area Address
  - #2 TLV: IS Reachability
  - #6 TLV: IS Neighbors
  - #24 TLV: IS Alias
- **Length**: 1-byte field containing the length of the payload data in the following field, Value.
- **Value**: 1-byte payload data, of the previously specified Type.

By using TLVs, the IS-IS protocol can also adapt to future requirements by introducing new TLV types, which is what it has been done through the years in order to adapt, for instance, new network layer protocols. TLVs make IS-IS a very extensible protocol, which is an important factor for its future evolution.

2.1.3 Information Flooding

The LSP flooding procedure in IS-IS is rather simple. When a LSP is originated in a router, the LSP is transmitted to all of its IS-IS neighbour routers that have an adjacency in the UP state. When a LSP is received, it is transmitted to all the neighbours with an adjacency in the UP state except from the neighbour that sent the LSP.

In Figure 2.1, all routers in the scenario belong to the same IS-IS area. Now let's focus in an LSP generated in router RA. Among other information, the LSP generated by router RA will advertise that it has UP state adjacencies with routers RB and RF, and will have the same sequence number. The black arrows represent this LSP flowing through the network. As it can be observed, the previously explained rule is followed in all the routers. For instance, in router RB, the LSP is received through the interface that links to router RA. What router RB does, is installing the

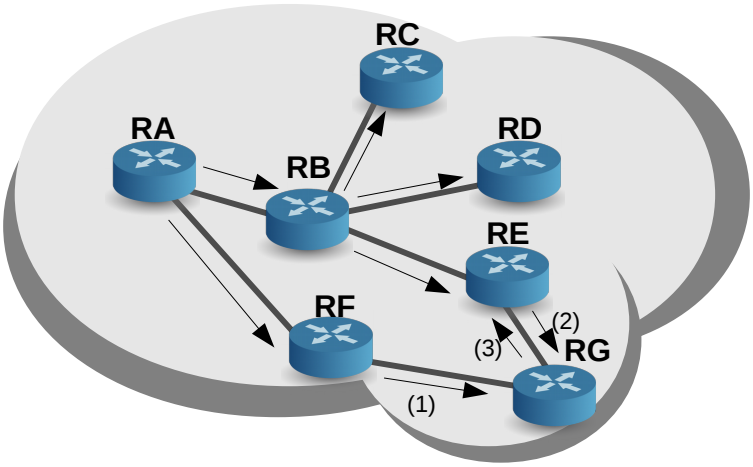


Figure 2.1: LSP flooding example

LSP information in its LSDB, and floods the LSP through the rest of interfaces. Now let's focus on routers RE, RF and RG. Router RG firstly receives the LSP from router RF, and installs it in its LSDB (1). Then, at the same time, routers RE and RG announce at the same time the same LSP to each other. What these two routers do, is checking the sequence number: Since RE and RG already had previously received the LSP with the same sequence number, it means that it does not provide any updated link-state information. So both LSPs (2 and 3) are discarded. This procedure avoids LSPs to circulate through the network endlessly.

2.1.4 Purging LSPs from the network

LSPs are used to update link-state and other types of information. LSPs can be also used to purge the information carried within an LSP from all the router's databases in the network. To achieve that, we could wait until the Lifetime of the LSP expires, which could take up to 65.535 seconds, or send a purge LSP.

To send a purge LSP, the router that originated the ordinary LSP to distribute information, sends an LSP with the Header and Checksum fields set to zero.

Purge LSP Usages

As well as when a router is removed from a network, purge LSPs are also used when a "special" router in a LAN (Designated Intermediate System) is re-elected. The function of a Designated Intermediate System (DIS) will be explained in the following section. Since the DIS floods information representing the LAN throughout the network, when the DIS is re-elected, or when it times out, the old DIS sends a purge LSP to withdrawn the previous information representing the network. The related LSP-ID information in the routers' databases will be removed 60 seconds after receiving the purge LSP. The LSP is kept in the databases 60 seconds after the purge to ensure that it is not relearned.

2.2 Pseudonodes

In a network containing several IS-IS routers, that maintain adjacencies with each other, an excessive LSP traffic may occur when a router joins or leaves the network. The problem is that as the LAN grows, the number of adjacencies among routers, and the number of LSP messages sent among each other is going to grow exponentially. Pseudonodes are used to solve this problem.

2.2.1 Pseudonode Concept

The function of the pseudonode is representing the LAN. That is, the LAN is represented as if it was another node in the network. However, the node that represents the LAN (pseudonode), is not a real physical node, as happens with routers.

The result of this concept of the network the following one:

- Each router is connected to the LAN (pseudonode).
- The pseudonode is connected to all routers.

By setting this model, the amount of adjacencies, and the number of LSP messages that are flooded throughout the network, does not grow exponentially.

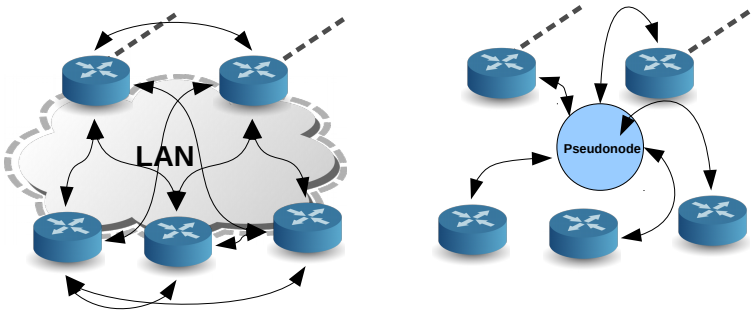


Figure 2.2: Pseudonode concept

In Figure 2.2, the pseudonode concept is shown. If pseudonode would not exist, the number of adjacencies among pairs of routers would be much greater. In this case, the number of adjacencies after applying the pseudonode concept becomes 5 versus 10. Note that the main problem in a growing network is not the database storage size. The main problem is the resource consumption generated from LSP flooding.

Pseudonode-ID Value in the LSP-ID field

As explained in Section 2.1, LSPs contain a 1-byte Pseudonode-ID within the LSP-ID field. When an LSP is generated by a router, representing a real node, the value of the Pseudonode-ID field will always be zero. On the other hand, when a DIS generates an LSP representing a pseudonode, the value will be different from zero. The System-ID value of the LSP-ID field is borrowed from the router that plays the DIS role.

Adjacencies Costs

When adjacencies are set among real routers and the pseudonode, different costs are set. On one hand, the cost for a pseudonode to reach a router will be always 0. On the other hand, the cost for a real router to reach the pseudonode will be a locally configured parameter.

2.2.2 Designated Intermediate System (DIS)

The pseudonode, has to be represented by a physical router, in order to apply the model and represent the LAN. The router that represents the pseudonode, as well as representing itself, is called the Designated Intermediate System (DIS). Every time a new router joins or leaves the network, will establish an adjacency with the DIS. This will just generate two new LSPs, one from the router that joins the network, and another one from the DIS.

Selecting the DIS

In order to select which router of the network is going to be the DIS, the following fields of the LAN IS-IS Hello (IIH) header are evaluated.

1. **Priority**: Comparing the local value for the Priority field, against the values received from other routers' IIHs. This value range is from 0 to 127, where 0 means not wishing to be the DIS at all.
2. **Source SNPA**: This value is equivalent to the sender's MAC address. This field is used to untie who is going to be the DIS, in the case that the Priority fields have the same value. The router that has the highest MAC address value is going to be the DIS.

As explained in the previous section, when a new DIS is selected, the old DIS router has to send purge LSP messages to update (remove) the old LAN information from the network routers' databases.

2.2.3 DIS Database Synchronization

In IS-IS, it is crucial to maintain the link state databases in all the topology routers synchronized. If the information in the databases is not updated, the effects might be, for instance, not reaching a destination or creating a routing loop.

To maintain databases synchronized and updated, IS-IS uses two types of packets:

- CSNP (Complete Sequence Number Packet)
- PSNP (Partial Sequence Number Packet)

The CSNP PDU is sent by the DIS, to all the routers in its LAN periodically. The purpose of this packet is sending a directory of its link state database to the other routers in the LAN. When the CSNP packet is sent by the DIS to the other routers in the LAN, the receiving routers check the LSP versions contained in the LSP.

If the CSNP PDU contains an older version of a LSP, the routers that have a more recent version, just flood the LSP to the DIS, so that the DIS updates the old LSP or group of LSPs.

If the CSNP PDU contains a more recent version of a LSP, the receiving router sends a PSNP PDU, asking the DIS to flood the updated version of the LSP.

In the case that a CSNP PDU lists a LSP not known by the receiving routers, the routers that do not have the LSP in their respective databases will send a PSNP PDU asking for that LSP.

2.3 Route Calculation

2.3.1 SPF Algorithm

From the topology information in the link state database, firstly, the SPF algorithm runs locally in each router. The SPF algorithm is based on a database that contains the node-to-node costs using three lists. From these information, the SPF algorithm determines the shortest path to each of the nodes.

Depending on the events that occur on the network, different passes in the SPF calculation are applied. The two different flavours are:

- Full SPF Run
- Partial SPF Run

The Full SPF Run is used to compute the topological grid, and also to recompute the reachable IP prefixes. The Partial SPF Run just computes leaf-related information of the topology.

Chapter 3

Scenarios

In the following Chapter, the IS-IS protocol is tested in different virtual scenarios. The scenarios are generated by using the VNX tool, and the IS-IS implementation by Quagga routing software.

3.1 Introductory Scenario

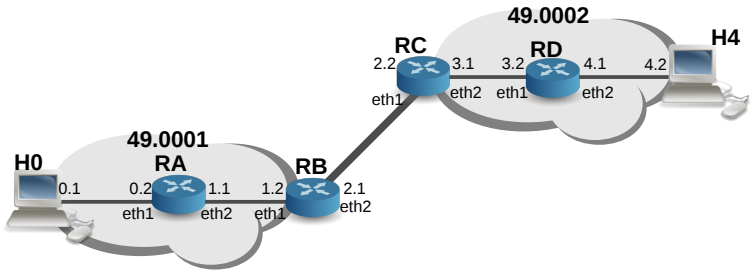


Figure 3.1: Introductory scenario

In Figure 3.1, routers and hosts have been assigned IP addresses from the 10.0.0.0/16 range. Networks are 10.0.X.0/24 in all cases (NetX). The desired configuration is setting two IS-IS areas (49.0001 and 49.0002), and a single Level-2 topology link/adjacency among routers RB and RC.

3.1.1 Starting the Scenario

Execute the following commands to initialize the scenario:

```
# vnx -f first-isis-scenario.xml -v --create
# vnx -f first_isis_scenario.xml -v --execute initial
```

3.1.2 Scenario Configuration

Setting Level-1 Topology Configuration

Once the scenario is initialized, the first two routers to be configured are RA and RB, within the same area, 49.0001, through a Level-1 topology link.

Firstly, routers' IP interfaces must be configured to work with IS-IS

```
ra# configure terminal
ra(config)# interface eth2
ra(config-if)# ip router isis 49.0001

rb# configure terminal
rb(config)# interface eth1
rb(config-if)# ip router isis 49.0001
```

The next step, is configuring the NET value in both routers, so that RA and RB can establish an adjacency. To do so, the following Quagga commands are used in router RA:

```
ra# configure terminal
ra(config)# router isis 49.0001
ra(config-router)# net 49.0001.0100.0000.1001.00
ra(config-router)# is-type level-1
```

The first line, is used to start the configuration mode. In the second line, enters to the IS-IS configuration, and at the same time, specifies the area. The third line specifies the topology level where the router will belong. The last command specifies the NET value of the router.

At this point, IS-IS Hello messages can be observed in Net1 using Wireshark. Since it has not been specified, router RA is sending Level-1 IS-IS Hello messages. The destination MAC addresses, are the broadcast MAC addresses for Level-1

The equivalent configuration is then applied in router RB:

```
rb# configure terminal
rb(config)# router isis 49.0001
rb(config-router)# net 49.0001.0100.0000.1002.00
```

In this case, there is not an explicit instruction to belong to Level-1 only. Since RB will be a Level-1 and Level-2 router, which is the default setting, it is not necessary to include the is-type command line.

The routers' adjacencies, and the Link State Database (for the two topology levels), can be checked by using this commands respectively:

```
ra# show isis neighbor
```

```
ra# show isis database
```

As expected, there is just one entry in the adjacencies table, and three LSP entries in the Link State Database. The entries in the database correspond to the two real nodes, plus the pseudonode.

The equivalent process is done in area 49.0002 with routers RC and RD. The command *show running-configuration* shows the resulting configuration, respectively:

```
. . .
!
interface eth2
 ip router isis 49.0002
 ipv6 nd suppress-ra
 no link-detect
!
. . .
!
router isis 49.0002
 net 49.0002.0100.0000.3001.00
 metric-style wide
!
. . .
```

```
 . . .
 !
interface eth1
 ip router isis 49.0002
 ipv6 nd suppress-ra
 isis circuit-type level-1
 no link-detect
!
. . .
!
router isis 49.0002
 net 49.0002.0100.0000.3002.00
 metric-style wide
 is-type level-1
!
. . .
```

Setting Level-2 Topology Configuration

The Level-2 Topology will be set in routers RB and RC, by enabling the two interfaces for IS-IS, that were not previously enabled:

```
rb# configure terminal
rb(config)# interface eth2
rb(config-if)# ip router isis 49.0001
```

```
rc# configure terminal
rc(config)# interface eth1
rc(config-if)# ip router isis 49.0002
```

At this point, Level-2 adjacency has been formed among routers RB and RC. The routing table in routers RB and RC, contain the 10.0.3.0/24 and 10.0.1.0/24 networks respectively. This is the expected result, since Level-1 link state topology is leaked in the Level-2 topology, but not in the opposite way.

However, a Quagga bug does not set the Attach Bit (ATT) in LSPs advertised by Level-1 Level-2 routers. Since the ATT is not set, the Level-1 routers in the areas, do not know which router/s in the area is Level-1 Level-2, so that traffic can be forwarded through those. The behaviour of Level-1 IS-IS routers is setting a default route among the routers that send LSPs with the ATT set. The result is that Level-1 routers cannot forward traffic to other IS-IS areas.

3.2 IS-IS/BGP Core Scenario

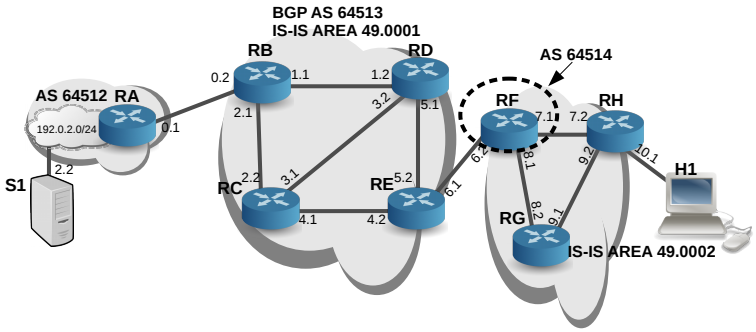


Figure 3.2: IS-IS/BGP Scenario

The purpose of the scenario in Figure 3.2 scenario is configuring the different parts of the network to establish connectivity among H1 and S1. The IS-IS Level-1 topology areas in the scenario are the following ones:

- Area 49.0001: Routers {RB, RC, RD, RE}
- Area 49.0002: Routers {RF, RG, RH}

Note that there is no Level-2 IS-IS topology connecting the previous two areas. This means that the same Area-ID could be used. However, different Area-IDs are used to better differentiate both.

Regarding BGP Autonomous Systems (AS), there is an iBGP Core, and two AS containing a single router:

- AS 64512: Router RA
- AS 64513: Routers {RB, RC, RD, RE}
- AS 64514: Router RF

The BGP prefixes that will be announced throughout the network, are the following ones:

- 192.0.2.0/24 from RA
- 10.0.7.0/22 from RF

3.2.1 IGP Configuration (IS-IS)

Firstly, IS-IS Area 49.0001 is configured, so that routers will be able to forward the prefixes announced using BGP. The System-ID value is inherited from the lowest IP address value within all the router's interfaces. The IS-IS configuration in routers is the following one:

- RB:

```
!
interface eth2
 ip router isis 49.0001
 ipv6 nd suppress-ra
 isis circuit-type level-1
 no link-detect
!
interface eth3
 ip router isis 49.0001
 ipv6 nd suppress-ra
 isis circuit-type level-1
```

```
 no link-detect
!
. . .
!
router isis 49.0001
 net 49.0001.0100.0000.0002.00
 metric-style wide
 is-type level-1
!
. . .
```

- RC:

```
. . .
!
interface eth1
 ip router isis 49.0001
 ipv6 nd suppress-ra
 isis circuit-type level-1
 no link-detect
!
interface eth2
 ip router isis 49.0001
 ipv6 nd suppress-ra
 isis circuit-type level-1
 no link-detect
!
interface eth3
 ip router isis 49.0001
 ipv6 nd suppress-ra
 isis circuit-type level-1
 no link-detect
!
. . .
!
router isis 49.0001
 net 49.0001.0100.0000.2002.00
 metric-style wide
 is-type level-1
!
```

- RD:

```
!
interface eth1
 ip router isis 49.0001
 ipv6 nd suppress-ra
 isis circuit-type level-1
 no link-detect
!
interface eth2
 ip router isis 49.0001
 ipv6 nd suppress-ra
 isis circuit-type level-1
 no link-detect
!
interface eth3
 ip router isis 49.0001
 ipv6 nd suppress-ra
 isis circuit-type level-1
 no link-detect
!
. . .
```

```
!
router isis 49.0001
 net 49.0001.0100.0000.1002.00
 metric-style wide
 is-type level-1
!
```

- RE:

```
. . .
!
interface eth1
 ip router isis 49.0001
 ipv6 nd suppress-ra
 isis circuit-type level-1
 no link-detect
!
interface eth2
 ip router isis 49.0001
 ipv6 nd suppress-ra
 isis circuit-type level-1
 no link-detect
!
. . .
!
router isis 49.0001
 net 49.0001.0100.0000.4002.00
 metric-style wide
 is-type level-1
!
. . .
```

At this point, all the IS-IS routers in area 49.0001, contain in its routing tables all the LANs that belong to the Level-1 topology. However, it is necessary to inject 10.0.0.0/30 and 10.0.6.1/30 in the area 49.0001 routers to correctly advertise the BGP prefixes. The route redistribution should be done in routers RB and RE respectively.

However, there is not an implementation in IS-IS to redistribute the connected routes, as there is in the other routing protocols implemented in Quagga. Since routes cannot be redistributed, the BGP next hop values will not be reached by routers, and traffic will not be forwarded. This issue limits the scenario development and the BGP settings in the routers.

Bibliography

[1] Hannes Gredler and Walter Goralski. *The Complete IS-IS Routing Protocol*. Springer Publishing Company, Incorporated, 2005.

[2] Cisco documentation. http://www.cisco.com/c/en/us/support/docs/ip/integrated-intermediate-system-to-intermediate-system-is-is/5739-tlvs-5739.html. Accessed: 2017-01-10.

[3] Quagga routing software suite. http://www.nongnu.org/quagga/. Accessed: 2017-01-10.